THREE-DIMENSIONAL HYDRODYNAMIC MODEL SET-UP: A CASE STUDY OF MIDDLE TENNESSEE RIVER

By

Shuvashish Roy

Dr. Jejal Bathi
Visiting Assistant Professor of Civil Engineering
(Committee Chair)

Dr. Arash Ghasemi
Director, Civil Infrastructure Laboratory
(Committee Member)

Dr. Weidong Wu
Associate Professor of Civil Engineering
(Committee Member)

Dr. Joseph Owino
Professor of Civil Engineering
(Committee Member)

Dr. Ignatius Fomunung
Professor of Civil Engineering
(Committee Member)

Dr. Mbakisya A. Onyango
Associate Professor of Civil Engineering
(Committee Member)

THREE-DIMENSIONAL HYDRODYNAMIC MODEL SET-UP: A CASE STUDY OF MIDDLE TENNESSEE RIVER

By

Shuvashish Roy

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial Fulfillment of the Requirements of the Degree of
Master of Science: Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May 2020

ABSTRACT

The Environmental Protection Agency's (EPA) Environmental Fluid Dynamics Code (EFDC) is a widely used hydrodynamics model that is no longer up to date with modern technology. As part of this research, an automated MATLAB based structured grid generator for EFDC was developed and was tested by developing a test model for Tennessee River near Chattanooga area. The test model was developed for a 9.92 km long segment of the Tennessee River starting from the downstream of the Chickamauga dam to upstream of Moccasin island. The model was calibrated against measured water flows, velocities and gage heights from January 1 to June 27, 2008. The Nash-Sutcliffe coefficients for velocities and gage heights were 0.788 and -6.61 respectively. Overall the model could simulate the field condition effectively. However, a detailed bathymetry is required, and the current limitation of the grid generators manual adjustments need to be addressed to produce better simulation results.

DEDICATION

This thesis is dedicated to my parents and my friends who have encouraged and believed in me for so many years.

ACKNOWLEDGEMENTS

I would like to thank Dr. Bathi for his guidance and for giving me the chance to work under his supervision. It has been a great honor for me to work with him. My sincere appreciation also goes to Dr. Arash Ghasemi and Mr. Babatunde Atolagbe for their collaboration in this research. This research was partially supported by the U.S. Geological Survey under Grant/Cooperative Agreement No. G16AP00084 and the Center of Excellence in Applied Computational Science and Engineering (CEACSE). Finally, I am thankful to my parents and friends who shared my enthusiasm and helped me go through my disappointments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EPA: The Environmental Protection Agency

EFDC: Environmental Fluid Dynamics Code

USGS: The United States Geological Survey

GIS: Geographic Information System

LIST OF SYMBOLS

b: Buoyancy

$\partial_z p$ = Excess hydrostatic pressure

ζ = vertical coordinates of the free surface

ρ = Density

CHAPTER I

INTRODUCTION

Significance

Use of numerical modeling to solve the intricate hydrodynamic processes in water problems can provide very useful information such as description of circulation, water levels, velocity, temperature variations and stratification processes and their effects on the transport of pollutants and water quality within a water body (Cedillo, 2015). Hydrodynamic models dealing with the mechanisms of flow to quantify the physical processes in water helps to understand the movement and transport of contaminants in water bodies and serves as a basis for water quality research (L. Liu, 2018). However, grid generation is an important and challenging preprocessing step for setting-up a detailed discrete hydrodynamic model for simulating river flows. Often times, resolution of the modeling study is determined by limitations of the grid generation programs. Also, compatibility of grid generators to match with a hydrodynamic model is another challenge. Environmental Protection Agency's Environmental Fluid Dynamics Code (EFDC) is one of the most popular models used for simulating surface water hydrodynamics and water quality. However, it's associated Fortran- based GEFDC grid generator for EFDC is complex and requires a substantial amount of mathematical knowledge and craftsmanship to set-up a compatible grid. Taking this into consideration, we developed a new algorithm and implemented in a MATLAB based grid generator program that automates reproducible grid generation from variety data source

formats. As part of this dissertation, we have presented the grid generation process using a real-world Tennessee River hydrodynamic model that was set up and executed to produce results and verify compatibility of our grid generator with EFDC model.

Objectives

The main objective of this study was to develop a user-friendly grid generator tool for one of the widely used hydrodynamic model EFDC and to demonstrate the newly developed grid generator with a test case model. The EFDC model set-up for a section of Tennessee River (TN River) was used as the test case model. This tool will set up the future opportunity for hydrodynamics, sediment transport and water quality of rivers and lakes studies that will have a great importance for water resources management.

The demonstrated case study model stretching from Chickamauga Dam to Maclellan Island incorporates flows from the two major incoming tributaries: North Chickamauga Creek and South Chickamauga Creek. The developed model was calibrated and validated by comparing with the observed flow, velocity and flow depth.

This thesis report is arranged such that Chapter II describes the existing approaches and application of hydrodynamic codes in different water bodies, grid generation techniques for hydrodynamic models and the governing equations for EFDC. Chapter III presents the grid generation process for EFDC using newly developed MATLAB program. Chapter IV discusses the hydrodynamic model set up and simulated results for Tennessee River. Chapter V is presented with the summary of the research outcomes and scope for future work.

CHAPTER II

REVIEW OF RELEVANT LITERATURE

Introduction

Hydrodynamic modeling of water bodies can be performed using one dimensional (1D), two dimensional (2D) or in three dimensional approaches. In 1D modeling, simplified equations of continuity and momentum are solved in one direction. These models assume small bottom slope and longer water lengths in comparison to the water depths. 1D models can simulate hydrodynamics (i.e. discharge and flow) only in the direction of the flow whereas, in 2-D modeling, equations of continuity and momentum are solved in two dimensions and results are calculated at each grid point in the solution domain. In 2D modeling approaches hydrodynamic information (i.e. water levels and discharge) is available at every grid point across the computational domain both in x and y directions. However, fine spatial resolution (dx) used in 2D models makes computing slower than the 1D models which requires a lot of computer memory (Ahmad & Simonovic, 2000). In 2D modeling approaches, an actual description of the bathymetry and topography is very crucial for prediction accuracy. Examples of 1D models include MIKE 11, HEC-RAS, and Infoworks RS whereas some prominent 2D models include MIKE 21(2D), TUFLOW, CE-QUAL-W2, GSSHA, DELF-FLS, HEC-RAS (2D) etc. Watershed Management System (WMS) a user-friendly watershed management platform provides a hydraulic interface making it compatible with HEC-RAS. The RAS model can be run as steady or unsteady state, and

results are used to delineate floodplain extents and animations of flood waves for complete 2D flood plain analysis (Bathi & Roy, 2020).

However, accurate representation of complex dynamics of a river may only be fully captured by the three-dimensional (3D) models that use Reynolds-averaged Navier stokes equations in finite differences, finite elements or finite volume techniques. In 3D hydrodynamic modeling the target water body is divided into computational cells both in horizontal and vertical dimensions, which enables an accurate description of complex surface water bodies and thus solves the three-dimensional governing equations (X, Y, Z momentum and continuity) in all the three directions. After 1960, many 3D hydrodynamic models have been developed using finite difference or finite volume equations. 3D hydrodynamic models, if coupled with water quality modules, provide the best supports for impact assessments and sustainable decision making. Though the set up and execution of a 3D model generally takes way more time than simpler 2D or 1D (quasi 2D) models additional time and costs are better justified with verified, more reliable and detailed simulated results. Besides, advancement with topographic data acquisition, high performance parallel processing 3D hydrodynamic modeling is being applied more in recent times. Examples of 3D hydrodynamic models include MIKE 21(3D), Delf3D, EFDC, MIKE 3 etc.

Hydrodynamics models that has been widely used in recent times are Curvilinear Hydrodynamics in Three Dimension, Z-grid version (CH3D-z), MIKE 3, Princeton Ocean Model (POM), Hydrologic Engineering Center River Analysis System (HEC-RAS) and EFDC (Cedillo, 2015).

MIKE 21 and MIKE 3 hydrodynamic models developed by DHI come with a large number of tools with user interfaces to setup the boundary conditions, bathymetry and other external forces for 2D or 3D hydrodynamic analysis. MIKE hydrodynamic module (HD) solves the equations for

the conservation of mass and momentum as well as for salinity and temperature in response to a variety of forcing functions. Both models provide flexible choice of rectangular, nested or flexible meshes with seamless coupled modeling. For high performance computing, it has parallelization techniques and modules to use graphical processing units (GPUs). MIKE HD modules can be used for lake and reservoir hydrodynamics, environmental impact assessments, coastal flooding and storm surge, inland flooding, overland flow and many other cases. MIKE 21 and MIKE 3 models were used to simulate hydrodynamics between Bay of Fundy and Salmon River (Marvin & Wilson, 2016). The developed model was calibrated, and the simulated results were in good agreement with the observed hydrodynamics. Another widely used hydrodynamic model, MIKE11, was set up and calibrated to simulate river stage hydrodynamics (Panda, Pramanik, & Bala, 2010) along with an artificial neural network model. Simulated results from MIKE 11HD showed good agreements with observed values. The Nash–Sutcliffe coefficient (NSE) and root mean square error (RMSE) values were 0.7836 and 1.00, respectively.

Princeton Ocean Model (POM), developed from the Blumberg Mellor model (Blumberg & Mellor, 1987), is a simple but powerful sigma coordinate and free surface ocean model with embedded turbulence, wave sub models and wet dry capability have been applied in many different water bodies including river estuaries. POM model was applied successfully in St. Andrew Bay and in Lake Michigan to simulate the circulation pattern of fresh water inflows (Beletsky & Schwab, 2001; Blumberg & Kim, 2000).

In 1990s and 2000s many other models including ECOM, NCOM, FVCOM etc. were developed from the widely popular POM model (Al-Zubaidi, 2016). A 3D numerical model based on the POM model with orthogonal curvilinear coordinate in the horizontal direction and sigma coordinate in the vertical direction has been developed and applied in Pear River estuary, China.

The verified and calibrated computation results represented field data with good accuracy (Chau & Jiang, 2001).

EFDC model has remained one of the most popular amongst researchers and has featured in many different applications since its development in 1992 (X. Liu, 2007). It uses both finite volume and finite difference techniques to solve the equations of motion. EFDC, developed by Dr. J. M Hamrick (1992), is an USEPA recommended, three-dimensional, continuous, advanced surface water model. This model is based on the continuity equation of fluid. It consists of four sub modules: 1) hydrodynamic module 2) water quality module 3) sediment transport module and 4) toxics module. It is a widely recognized simulation platform with a multi-tasking, highly integrated modular computational fluid dynamics package that can be used for understanding and predicting the environmental fluid flows with transportation and mixing associated dissolved or suspended materials, as well as for modeling pollutants and pathogenic organism transport from point and non-point sources (Cunanan & Salvacion, 2016; Wang et al., 2014). This three-dimensional surface water modeling system is normally used for hydrodynamic and reactive transport simulations of rivers, lakes, reservoirs, wetland systems, estuaries, and the coastal ocean (Cunanan & Salvacion, 2016; J. M. Hamrick & Mills, 2000). It is an open source ,public domain (Cedillo, 2015; Cunanan & Salvacion, 2016) model used widely by universities, governmental agencies and engineering consultants within and outside the USA. However, it was maintained and developed by Tetra Tech Inc. with primary support from the United States Environmental Protection Agency (Cunanan & Salvacion, 2016; JM Hamrick, 2002) and lacks up-to-date technology.

Using horizontal orthogonal and sigma vertical coordinates, this model is capable of cohesive & non-cohesive sediment transport, water quality, near & far field discharge dilution

from point and nonpoint sources, eutrophication, toxic chemicals fate &transport modeling by linking with its hydrodynamic module. To avoid staircase grids in case of irregular bathymetry the whole water column is divided into the same number of layers across the water body using sigma coordinate transformation for smooth representation of topography. For grid generation it has a Fortran based GEFDC program though it requires a substantial amount of knowledge and craftsmanship to understand the mathematics of complex grid generation processes. EFDC has been used for more than 80 modeling studies of rivers, lakes, estuaries, coastal regions and wetlands in the United States and abroad including governmental agencies, universities and engineering consultants. For this study EFDC has been chosen because of its 3D simulation capability, opportunity to develop new user-friendly tools, and public domain availability.

In this section some successful modeling studies performed using the EFDC model are discussed.

Devkota and Fang (2015) developed a 17-km long EFDC hydrodynamic model for Mobile River, Alabama. Inflow from upstream, downstream tides were used as boundary conditions. The model was calibrated for water levels and velocity profiles for a period of over three months. The Nash-Sutcliffe (NS) coefficients for water levels were greater than 0.94 and for water temperatures ranged from 0.88 to 0.99 which represents the model's capability to capture the real-world variability of the hydrodynamics of the river.

Ji et al., (2000) developed a 3D hydrodynamic model for sediment transport model for Morro Bay, CA to compare the locations of sediment transport with the historical results. Though the modeling results showed good agreement with that of the observed historical results for over 31 days, during the six months of the simulation period model validation and calibration were not reported.

Jin et al., (2002) developed a three-dimensional EFDC hydrodynamic model with statistical validation for Lake Okeechobee, FL. The statistical analyses i.e. mean error, mean absolute error, root-mean-square error (RMSE), maximum absolute error (MAE), and the relative RMSE were evaluated for the model validation which showed that the simulated water surface elevations were in good agreement with the mean absolute errors ranging from 0.01 to 0.02 m, and the RMSE ranging from 0.012 to 0.027 m (Jin et al., 2002). The average absolute value of the relative errors at all stations was 1.4 cm, and the average relative RMSE was 6.89%.

Sucsy et al. (2002) measured and simulated times series of water surface elevation for St. Johns river with model calibration and validation. R-squared (R^2) value, RMSE, average absolute error (AVAE), Nash-Sutcliffe coefficient (NS) calculated in this study demonstrated the model's capability to correctly simulate the measured longitudinal water level variation. To capture the hydrodynamic variables over a wide range of weather conditions, model validation was performed for a ten-year time period so that simulated results can be used for forecasting altered future conditions including extreme scenarios, i.e. draught or flooding events.

EFDC Governing Equations

The formulation of the governing equations for ambient environmental flows characterized by horizontal length scales which are orders of magnitude greater than their vertical length scales begins with the vertically hydrostatic, boundary layer form of the turbulent equations of motion for an incompressible, variable density fluid. To accommodate realistic horizontal boundaries, it is convenient to formulate the equations such that the horizontal coordinates, x and y, are curvilinear and orthogonal. To provide uniform resolution in the vertical direction, aligned with the gravitational vector and bounded by bottom topography and a free surface permitting long

wave motion, a time variable mapping or stretching transformation is desirable. The mapping or stretching is given by:

$$z = (z^* + h)/(\zeta + h) \quad (1)$$

where z^* denotes the original physical vertical coordinates

h denotes vertical coordinates of bottom topography

ζ: vertical coordinates of the free surface

Transforming the vertically hydrostatic boundary layer form of the turbulent equations of motion and utilizing the Boussinesq approximation for variable density results in the momentum and continuity equations and the transport equations for salinity and temperature in the following form:

$$\partial_t(mHu) + \partial_x\left(m_y Huu\right) + \partial_y(m_x Hvu) + \partial_z(mwu) - (mf + v\partial_x m_y - u\partial_y m_x)Hv =$$

$$-m_y H\partial_x(g\zeta + p) - m_y(\partial_x h - z\partial_x H)\partial_z p + \partial_z(mH^{-1}A_v\partial_z u) + Q_u \quad (2)$$

$$\partial_t(mHv) + \partial_x\left(m_y Huv\right) + \partial_y(m_x Hvv) + \partial_z(mwv) + (mf + v\partial_x m_y - u\partial_y m_x)Hu =$$

$$-m_x H\partial_y(g\zeta + p) - m_x\left(\partial_y h - z\partial_y H\right)\partial_z p + \partial_z(mH^{-1}A_v\partial_z v) + Q_v \quad (3)$$

In these equations, u and v are the horizontal velocity components in the curvilinear, orthogonal coordinates x and y.

m_x and m_y are the square roots of the diagonal components of the metric tensor, $m = m_x m_x$ is the Jacobian or square root of the metric tensor determinant.

f= Coriolis parameter

A_v= Eddy Viscosity (vertical turbulent)

Q_u and Q_v= Moment source-sink terms

H $= \zeta + h$ is the total depth, where $w^* = 0$

Density ρ is a function of temperature (T), and salinity (S). The buoyancy is defined by the following equations:

$$\partial_z p = -gH(\rho + \rho_0)\rho_0{}^{-1} = gHb \quad (4)$$

where,

$\partial_z p$ = Excess hydrostatic pressure

ρ = Density (depends on temperature T and salinity of water S)

b= Buoyancy

The continuity equation is defined by:

$$\partial_t(m\zeta) + \partial_x(m_y Hu) + \partial_y(m_x Hv) + \partial_z(mw) = 0 \quad (5)$$

$$\partial_t(m\zeta) + \partial_x(m_y H \int_0^1 u\, dz) + \partial_y\left(m_x H \int_0^1 v\, dz\right) = 0 \quad (6)$$

The boundary condition used for the above equation: w=0 at Z= (0,1)

$$\rho = \rho(p, S, T) \quad (7)$$

The following equations are used for salinity and temperature respectively:

$$\partial_t(mHS) + \partial_x\left(m_y HuS\right) + \partial_y(m_x HvS) + \partial_z(mwS) = \partial_z(mH^{-1}A_b\partial_z S) + Q_s \quad (8)$$

$$\partial_t(mHT) + \partial_x\left(m_y HuT\right) + \partial_y(m_x HvT) + \partial_z(mwT) = \partial_z(mH^{-1}A_b\partial_z T) + Q_T \quad (9)$$

Where,

Q_s and Q_t are the source and sink terms that include subgrid scale horizontal diffusion and thermal source and sinks.

A_b: Vertical turbulent diffusivity

The vertical velocity, with physical units, in the stretched, dimensionless vertical coordinate z is w, and is related to the physical vertical velocity w^* by:

$$w = w^* - z(\partial_t\zeta + um_x{}^{-1}\partial_x\zeta + vm_y{}^{-1}\partial_y\zeta) + (1-z)(um_x{}^{-1}\partial_x h + vm_y{}^{-1}\partial_y h) \quad (10)$$

The total depth, $H = \zeta + h$, is the sum of the depth below and the free surface displacement relative to the undisturbed physical vertical coordinate origin, $z^* = 0$.

The pressure p is the physical pressure in excess of the reference density hydrostatic pressure, $\rho_0 g H\ (1 - z)$, divided by the reference density, ρ_0.

In the momentum equations (2, 3) f is the Coriolis parameter, A_v is the vertical turbulent or eddy viscosity, and Q_u and Q_v are momentum source-sink terms.

$$A_v = \varphi_v q l = 0.4\left(1 + 36R_q\right)^{-1}\left(1 + 8R_q\right)ql \tag{11}$$

$$A_b = \varphi_b q l = 0.5\left(1 + 36R_q\right)^{-1} ql \tag{12}$$

$$R_q = \frac{gH\partial_z b}{q^2}\frac{I^2}{H^2}$$

where the so-called stability functions φ_v and φ_b account for reduced and enhanced vertical mixing or transport in stable and unstable vertically density stratified environments, respectively. The turbulence intensity and the turbulence length scale are determined by a pair of transport equations:

$$\partial_t(mHq^2) + \partial_x\left(m_y Huq^2\right) + \partial_y\left(m_y Hvq^2\right) + \partial_z(mHwq^2) = \partial_z(mH^{-1}A_b\partial_z q^2) +$$

$$Q_q + 2mH^{-1}A_v((\partial_z u)^2 + (\partial_z v)^2) + 2mgA_b\partial_z b - 2mH(B_1 l)^{-1}q^3 \tag{13}$$

$$\partial_t(mHq^2 l) + \partial_x\left(m_y Huq^2 l\right) + \partial_y\left(m_y Hvq^2 l\right) + \partial_z(mwq^2 l) = \partial_z(mH^{-1}A_b\partial_z q^2 l) +$$

$$Q_l + mH^{-1}E_1 l A_v((\partial_z u)^2 + (\partial_z v)^2) + mgE_1E_3 l A_b\partial_z b - mHB_1{}^{-1}q^3(1+E_2(\kappa L)^{-2}l^2) \tag{14}$$

$$L^{-1} = H^{-1}(z^{-1} + (1 - z)^{-1}) \tag{15}$$

where B_1, E_1, E_2, and E_3 are empirical constants and Q_q and Q_l are additional source-sink term such as subgrid scale horizontal diffusion. The vertical diffusivity, A_q, is in general taken equal to the vertical turbulent viscosity, A_v.

Detailed discussion of the EFDC theory and computation can be found at EFDC hydrodynamic and mass transport and user manual (JM Hamrick, 2002; JM Hamrick, 2007).

Grid Generation in Hydrodynamic modelling

As with any numerical simulation solver, hydrodynamic model involves three stages including the pre-processing phase which includes development and organization of input data such as grids, processing phase where the main simulation occurs, and post-processing phase which basically involves visualization of the outputs of the simulation. Thus, the role of grid generation is also very crucial to the realization of accurate simulation of the problem.

Creating and assessing a suitable grid generation program for EFDC model has been quite challenging. In fact, according to Tetra Tech Inc, the maintainer of EFDC, existing grid generation software generally requires a lots of user experience and artistry to support the mathematical grid generation program(Xiong, 2010). Moreover, existing tools can in certain cases be erroneous or even expensive for personal or non-commercial use.

A popular grid generation software for EFDC is GEFDC grid generation program. This program is a FORTRAN code which is developed and capable of producing structured rectangular and curvilinear meshes (Alarcon, McAnally, & Pathak, 2012; Tetra Tech Inc, 2002) and it was originally designed by Tetra Tech Inc as the complementary grid generation tool for EFDC models. However, the GEFDC usually requires modification to suit the model and the success of this depends on the level of skills and experience of the user (Xiong, 2010).

Another existing grid generation, the visual orthogonal grid generation (VOGG), involves a FORTRAN implementation of a novel physical domain grid generation algorithm, a Windows/ GIS based interface for creating necessary input files and displaying output results, and several

utility programs (Xiong, 2010). It was developed as a component of the United States Environmental Protection Agency (US EPA) Region 4 Total Maximum Daily Load (TMDL) modeling toolbox and specifically supports curvilinear-orthogonal grid generation for the Environmental Fluid Dynamics Code (EFDC) (Xiong, 2010). But, a variety of ASCII text output files are created which readily allow grid information to be processed and reformatted for other hydrodynamic and transport models employing both orthogonal and non-orthogonal curvilinear grid formulations (Tetra Tech Inc 2002). Although the VOGG is popularly known and adopted, it is considered unstable and errors occur often according to Xiong (2010).

Another grid generation software that exists, and in fact, has been a more dominantly used is the EFDC Explorer. Created by Dynamic Solutions LLC, the program has a user interface and a grid generation tool (Xiong, 2010) thus making it simple and easier for users to work with. However, this tool is a commercial software and may not always be affordable to non-commercial (such as academic researchers and public agencies) users.

Apart from developing computational grids for EFDC models, grids can generally be created for other models as necessary since other modeling platforms exist that are different from EFDC. Schubert, Sanders, Smith, and Wright (2008) developed unstructured mesh generation framework for hydrodynamic modeling for urban flooding. Their methodology focuses on strategies for effective integration of geospatial data for unstructured mesh generation, building representation and flow resistance parameterization. Mesh generation was done using Triangle to generate Delaunay triangle meshes with inputs to Triangle including a polygon defining the outer boundary of the domain being considered, additional polygons defining interior boundaries called mesh holes, polylines that fix break-lines in the mesh, and points that fix vertices. In the development of the meshes, three data sources were utilized. These include Light Detection and

Ranging (LiDAR) terrain height surveys, aerial imagery, and vector datasets which includes the building footprint polygons. Three unique unstructured meshing techniques were developed, namely building-hole method (BH), building-block method (BB), and no-building method (NB), all according to the authors can be viewed as three options for urban flooding with different pre-processing demands. In the building-hole method, the first step is the outer boundary definition, from LiDAR data, that follows the course of the river such that is extended beyond the floodplain, to avoid interference between boundary and actual river flow. Thereafter, the interior hole definition is done by extracting location and shape of buildings from Digital Surface Models (DSMs) or aerial photography and converted to vector data structures or polygons using GIS software or platform. With the interior and exterior boundary defined, the data are used as input by Triangle to generate the required meshes.

Finally, a Digital Terrain Model (DTM) is used to interpolate terrain heights at mesh nodes or vertices. For building-block (BB) method, only exterior boundary is considered as input to Triangle but the building footprints and heights (or blocks) are burnt into the Digital Terrain Models (DTM) prior to mesh interpolation. Similarly, in no-building (NB) method, only the exterior boundary data is considered for mesh generation while a bare-earth Digital Terrain Model (DTM)is considered for mesh interpolation. The BH method is mostly used while NB is used least. To complement the capabilities of Triangle (Shewchuk 1996) for mesh generation, building shape data is output to a CAD Digital Exchange Format (DXF) and a utility is created to convert the created ASCII DXF file into ASCII format required by Triangle. Similarly, in cases where vector datasets of the buildings are not available, Digital Surface Models can be processed to represent building footprints followed by preparation a similar conversion utility for input to Triangle. Figure 1 depicts the meshing framework described by the authors.

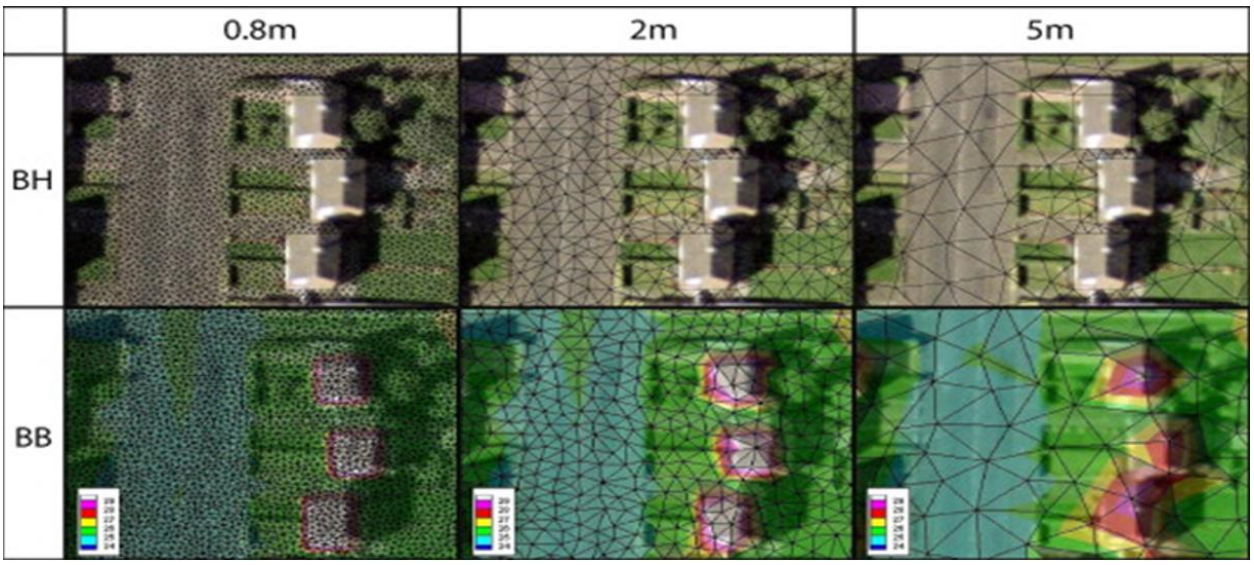

Figure 1 Meshes based on building-hole (BH) and building-block (BB) approaches (Schubert et al. 2008)

The aim is study to utilize existing mesh generation techniques and tools to develop algorithm that can be implemented in programs such as MATLAB for automatically generating mesh representation of any arbitrary waterscape using cheaply available resources including geospatial data and Triangle (Shewchuk 1996) such that the meshes can then be used for relevant applications like computational environmental fluids dynamics analysis and alleviates the difficulties inherent in existing methods.

CHAPTER III

GRID GENERATION

Grid-Generation Process

The grid generation process consists of seven major steps which can be simply re-categorized into four combinations. This first combination involves getting geospatial data from a suitable repository and extract coordinates of the points bounding the waterscape of interest. Following this phase, triangular grids are created for the wet areas and subsequent quadrilateral grids are generated for the entire waterscape (i.e. both wet and dry areas of the domain) using the triangular grids. In the third phase, the cells are assigned with tags as recommended for the EFDC model. Lastly, the resulting grids are written in an output for visualization and then further processed as input files for the EFDC model. Figure 2 depicts the flow chart for the procedure and Figure 3(a) and 3(b) present the major part of the algorithm.

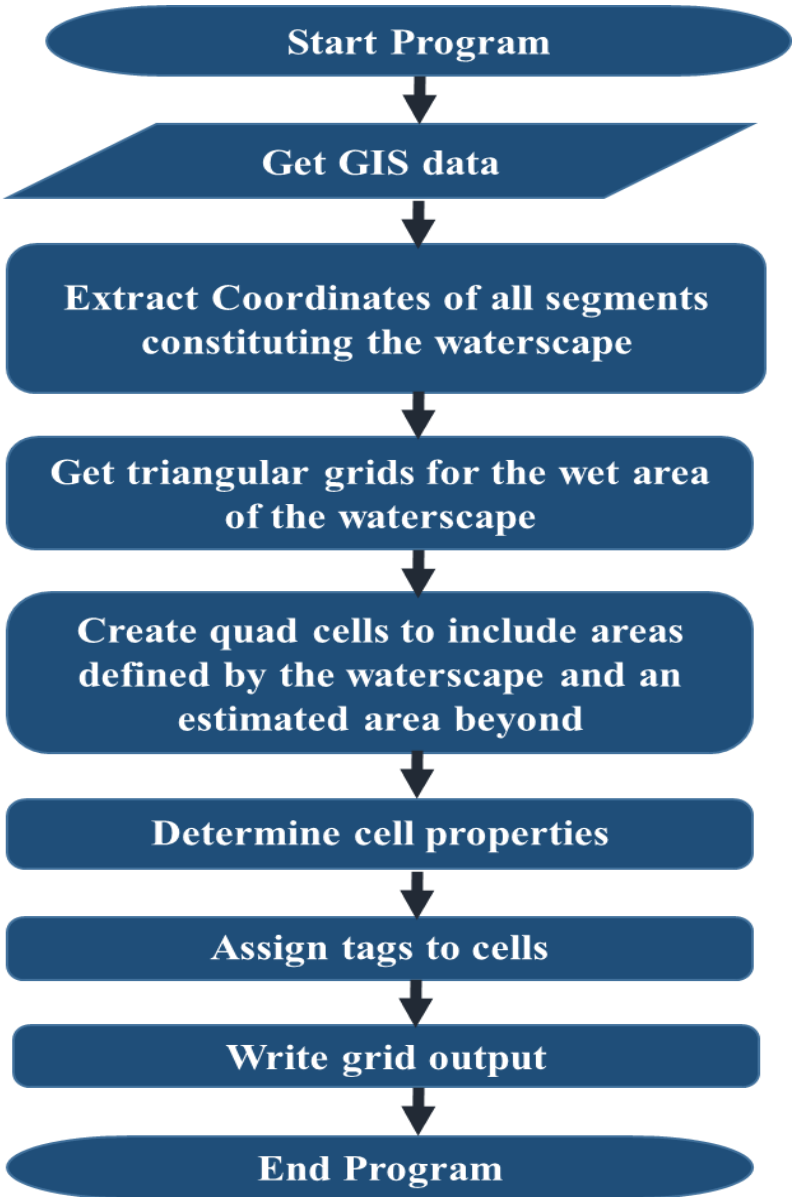

Figure 2 A flowchart for the grid generation for EFDC

The following is the algorithmic process of the framework (Roy, Atolagbe, Ghasemi, & Bathi, 2020). More details on this algorithm can be found in (Atolagbe, 2019).

Algorithm RiverGrid

This function generates the grid data for the Waterscape mesh
Input: nx, ny,N_{seg},$XYZRaw$, $EdgesRaw$
Output: $cells$

Step 1: Get triangular grids for Waterscape
Extract coordinates and connectivity into arrays XYZ, $Edges$ respectively.
for i in N_{seg} **do**
    Store $XYZ_{seg}, Edges_{seg}$ in $XYZ, Edges$ ▷ Input to **WaterscapeTriangulation**
end for
if $N_{seg} = 1$ **then**
    Island doesn't exists
    $Holes \leftarrow null$
else
    Let $i_{seg} \leftarrow 1$
    for i_{seg} in $(N_{seg} - 1)$ **do**
        Extract $XYZ_{seg}, Edges_{seg}$ ▷ Input to **GetIslandTri**
        $XYZ_{island}, ICON_{island} \leftarrow$ **GetIslandTri**
        Pick any triangle
        $r \leftarrow$ random cell (or row) number in $ICON_{island}$
        $RndmCell \leftarrow ICON_{island}(r)$ ▷ RndmCell is a $1x3$ vector
        $pt_i = RndmCell(i)$
        $V_j \leftarrow XYZ_{island}(pt_j, 1 : end); j = 1 : 3$ ▷ V is $1x3$ vector of coordinates of the vertices pt_i.
        $x_k = V_j(k, 1), k = 1 : 3$
        $y_k = V_j(k, 2), k = 1 : 3$
        $Holes(i) \leftarrow \frac{\sum_{k=1}^{3} x_k, y_k}{3}$ ▷ Input to **WaterscapeTriangulation**
    end for
end if
$XYZ_{tri}, ICON_{tri} \leftarrow$ **WaterscapeTriangulation**

Step2: Define quad cells beyond the Waterscape domain
Extend two distant corner points defined by XYZ_{tri} to $P1, P2$ by a desired length
Let $x1 = P1_x$, $y1 = P1_y$,$x2 = P2_x$, $y2 = P2_y$,
Calculate cell size: $dx, dy \leftarrow \frac{x2, y2 - x1, y1}{nx, ny}$

Figure 3(a) Pseudocode of grid generation algorithm (part 1)

Algorithm RiverGrid(continued)

Hold $x1_{init} \leftarrow x1$
Hold $y1_{init} \leftarrow y1$
Get coordinates of grid cells
initialize $inode, jnode \leftarrow 1$
initialize $jj \leftarrow 1$
for $jnode$ in $(ny + 1)$ **do**
    for $inode$ in $(nx + 1)$ **do**
        $x(inode, jnode) = x1 + dx$;
        $y(inode, jnode) = y1$;
        Check point x, y in $ICON_{tri}$; Add tags $T = 1$ or 0
        $T(inode, jnode) \leftarrow$ **CheckPointInCell**
        $index(inode, jnode) = jj$
        $jj = jj + 1$
        $x1 = x1 + dx$
    end for
    $x1 = x1_{init}$
    $y1 = y1 + dy$
end for

Calculate cell values
Initialize $icell, jcell \leftarrow 1$
for $jcell$ in ny **do**
    for $icell$ in nx **do**
        $cells\{icell, jcell\}.corners.x = x(icell, jcell, x(icell + 1, jcell), x(icell + 1, jcell + 1), x(icell, jcell + 1)$
        $cells\{icell, jcell\}.corners.y = y(icell, jcell, y(icell + 1, jcell), y(icell + 1, jcell + 1), y(icell, jcell + 1)$
        $cells\{icell, jcell\}.centroid = \bar{x}, \bar{y}$
        $cells\{icell, jcell\}.dxdy = dx, dy$
        $cells\{icell, jcell\}.tags = T(icell, jcell), T(icell + 1, jcell, T(icell + 1, jcell + 1), T(icell, jcell + 1)$
    end for
end for

Step3: Write EFDC File
$cells \leftarrow$ **CreateEFDCTags**
Output grd
End

Figure 3(b) Pseudocode of grid generation algorithm (part 2)

GIS Data Acquisition and Coordinates Extraction

Although the grids needed for EFDC are quadrilateral, the triangle grids here are only needed to serve as basis for establishing cell nodes that fall within the wet area. This unstructured triangular grid structure, as shown in Figure 4, is generated using Triangle (Shewchuk, 1996). The

quadrilateral cells on the other hand are generated by dividing the entire waterscape (including wet and dry areas) into a finite number of equal square cells as shown in Figure 5.

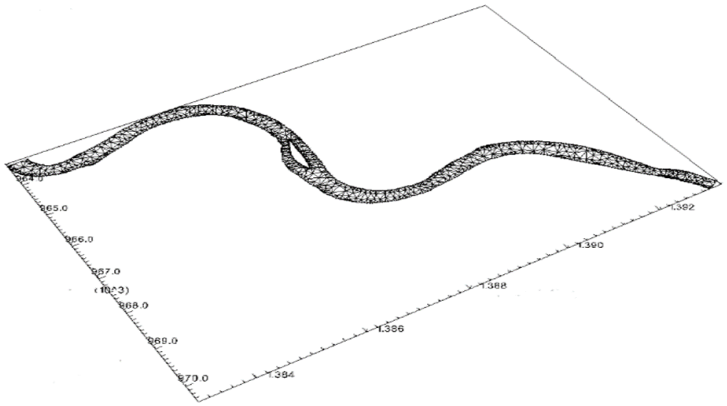

Figure 4 Triangular grids for wet area of the water scape

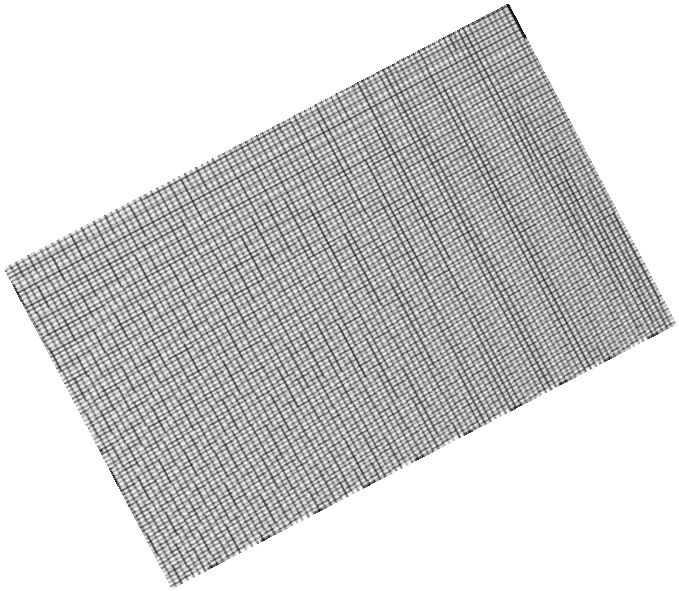

Figure 5 Quadrilateral cells covering the entire scape

Determining cell properties and assigning cell tags

In this phase, the properties of the quadrilateral cells are determined in a structured grid pattern. These values include cell indices, coordinates of cell vertices, coordinate of centroids and cell sizes (*dx, dy*). Once the points (or nodes) are generated, they are checked whether they fall inside the wet area (i.e. within the unstructured triangular grid) or dry area with a value of 1 or 0 assigned respectively to the nodes. This is more like super-imposing the triangular grids over the quadrilateral cells as shown in Figure 6.

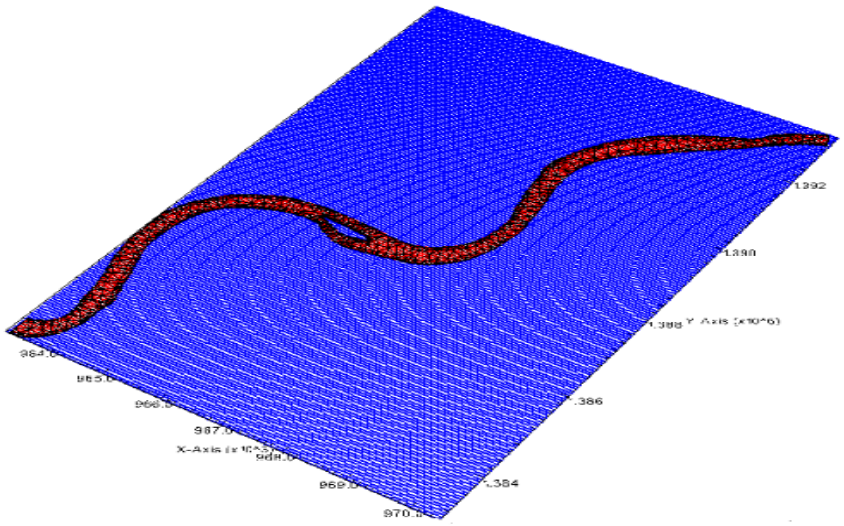

Figure 6 Superposition of triangular grids and quadrilateral cells

It is done to extract points that fall in the wet area of the domain. The values are then used to determine the tags of the cell. The tags are assigned to the cells according to the definitions of EFDC model as shown in Figure 7 & 8. Depending on the structure of these nodal tags, the square cells are assigned with values of 0, 1, 2, 3, 4, 5 and 9 which are required for the EFDC model.

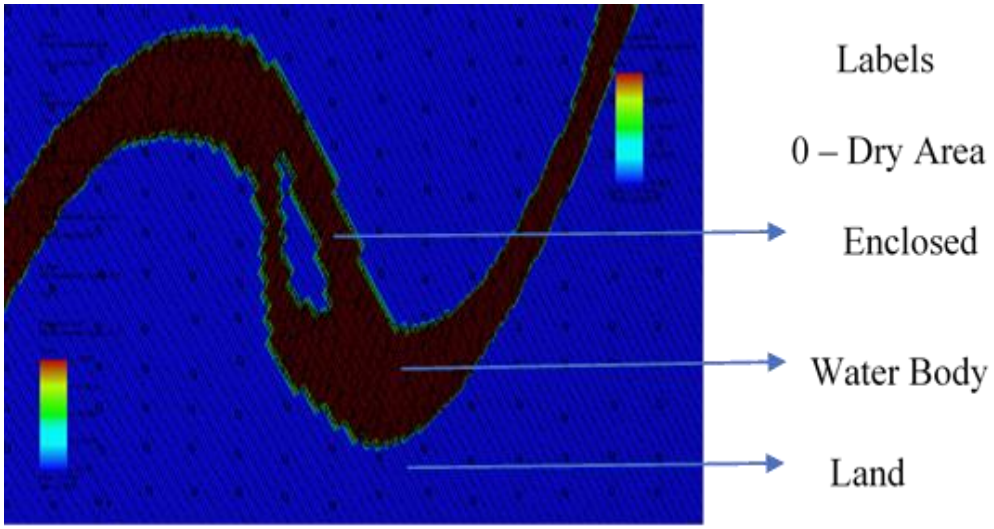

Figure 7 Blown-out section of the meshed waterscape

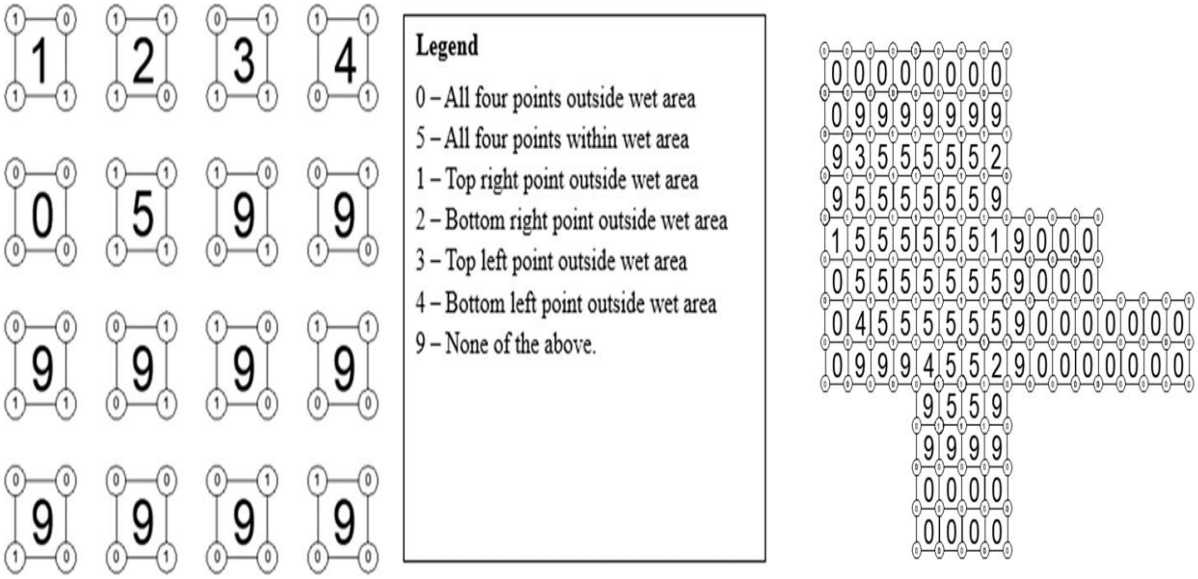

Figure 8 cell tags based on nodal tags

Writing Grid output

These EFDC tags, and other properties of the cells including cell indices (i, j), coordinates of cells corner points, coordinates of centroids of the cells, and length of cells in both directions

defined as (dx, dy) are output as a ".dat" file ready to use in EFDC model. Figure 2 presents a simple flowchart of the procedure. This .dat file is further processed to generate three input files including "cell.inp" (Figure 9), "lxly.inp", and "dxdy.inp" – which are required for the EFDC model. However, the generated grid needed some manual adjustments for the lxly and dxdy input files for the compatibility of EFDC.

Figure 9 Cell.INP input file for EFDC model

.

CHAPTER IV

TENNESSEE RIVER HYDRODYNAMIC MODEL

Model Description

To test the compatibility of the grid generator for EFDC, a hydrodynamic model of the Tennessee River near Chattanooga, TN is developed as a case study (Figure 10). Modeling of Tennessee River near the downtown region is crucial for assessing the risk of urban flooding during an extreme event as well as its water quality issues.

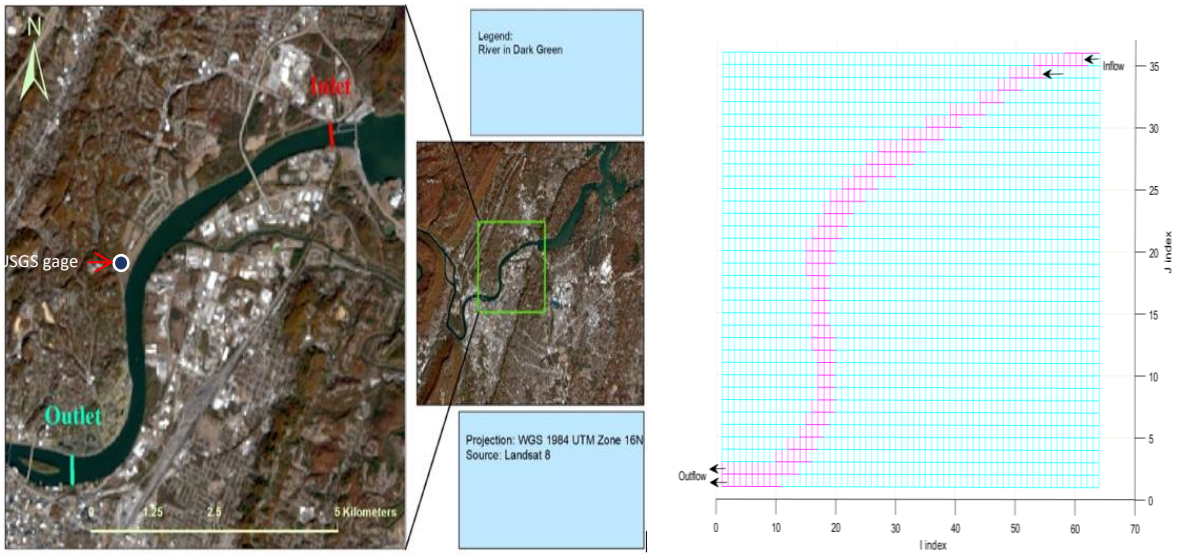

Figure 10 Tennessee River segment and grid for the hydrodynamic model

Therefore, the model segment starting from downstream of the Chickamauga dam to upstream of Moccasin island (Figure 10) is chosen for this research. The 2.41 km^2, 9.92 km long (6.2 mile) river model consists of 382 rectangular grid cells with an average dimension of 157m by 80.1m distributed in 42 rows and 70 columns in two vertical layers. The US Army Corps of Engineers 2000 map is used for the model bathymetry. The coordinate system chosen for the model was WGS 84 UTM zone 16N and the units were in meters. Details about the model information and cell statistics are shown below in Table 1 and Table 2.

Table 1

Model Information

Model Start Time	1/1/2008
Map Projection	WGS 1984 UTM Zone 16N
Conversion Factor	1
Maximum Grid Column (IC)	70
Maximum Grid Row (JC)	42
Number of Layers	2
Total Number of Cells	382

Table 2

Cell statistics

Parameter	Number of Cells	Maximum Value	Minimum Value
Dx (m)	382	80.2	80.1
Dy (m)	382	158	157
Roughness	382	0.025	0.025
Vegetation	382	0	0

Boundary conditions

Boundary conditions needed to be specified for the hydrodynamic model set up for the Tennessee River using EFDC. The time series values of flow sources were uniformly distributed in the vertical direction at the upstream boundary location cells (shown as inlet/inflow in Figure 10). Water pressure time series was used as the downstream boundary condition which was tuned to get desired flow results during the calibration process. The USGS flow station 035680000 at Chattanooga was located in the upper section of the river that accounted for both South and North Chickamauga tributaries. Flows from this station were used as flow data source. Though the location of the station was below the upstream boundary location, it was still close to the upstream boundary and was expected to represent the flows at the upstream boundary. Also, this provided the opportunity to compare the simulated flows at the USGS station while they were assigned at the upstream boundary.

Figure 11 shows summary of measured flow at the USGS station that was assigned as the upstream boundary condition during the model setup whereas Figure 12 presents the gage heights at the same USGS station. These measured results were later compared with the simulated results from the model at cell (27,28) because of its close proximity to the USGS station. Further details of the gage height and flow comparison to that of the simulated flows are discussed under 'Calibration' section of this chapter.

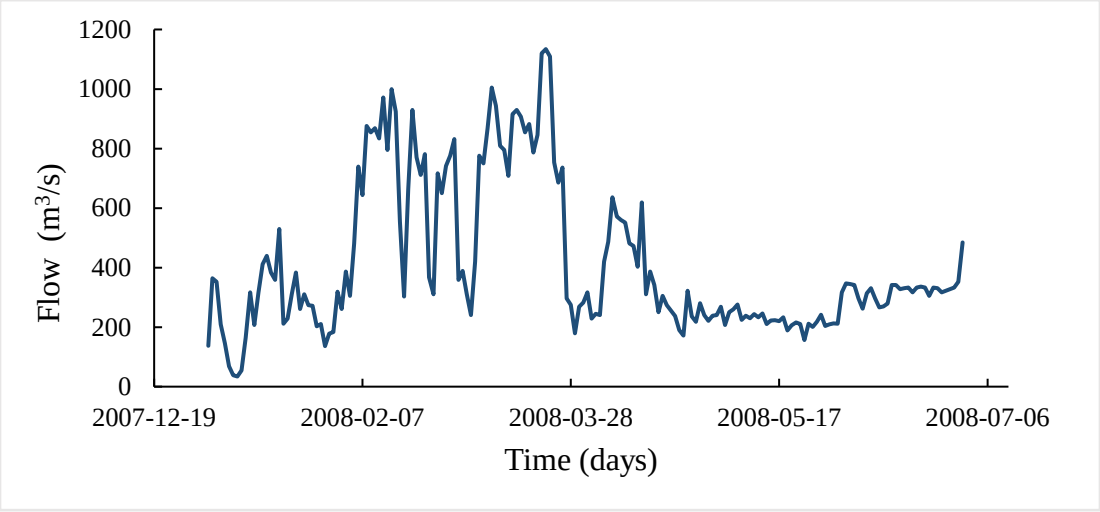

Figure 11 Inflow at USGS 03568000 Tennessee River, Chattanooga

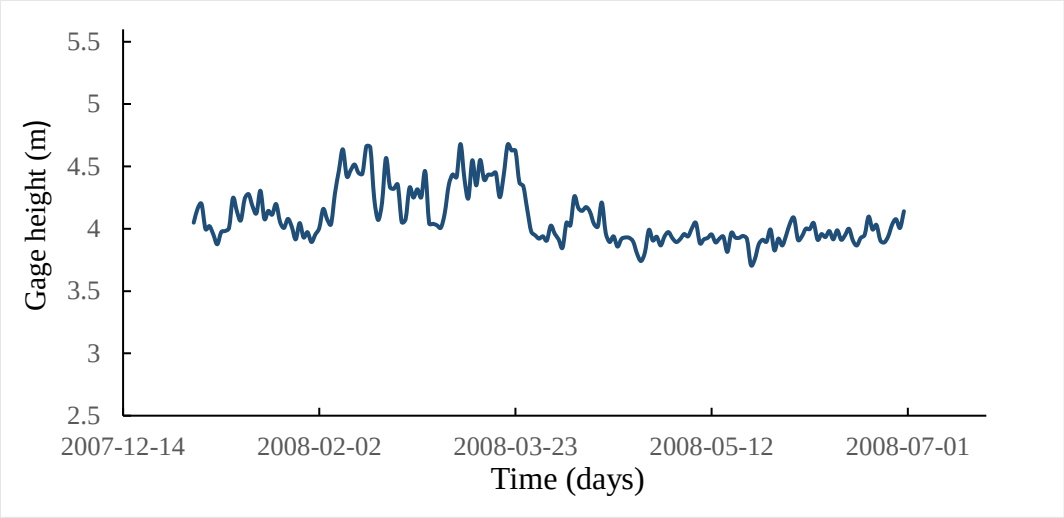

Figure 12 Gage height at USGS 03568000 Tennessee River, Chattanooga

Bathymetry

For smooth bathymetry data input the whole river segment was divided into four equal parts along the length. Each part was further subdivided into several equal segments along the width and length of the river according to the number of cells in that specific section. Bottom elevation was linearly distributed from upstream to downstream and from the bank of the river to the middle part of each segment. The bathymetry of the river took more like a trapezoidal shape because of very few available contour lines from the Army Corps Map used for the bathymetry and relatively few numbers of grid cells along the width of the river. Figure 13 and Figure 14 show the 3D bottom elevation and 2D contour plot of the TN River bathymetry respectively.

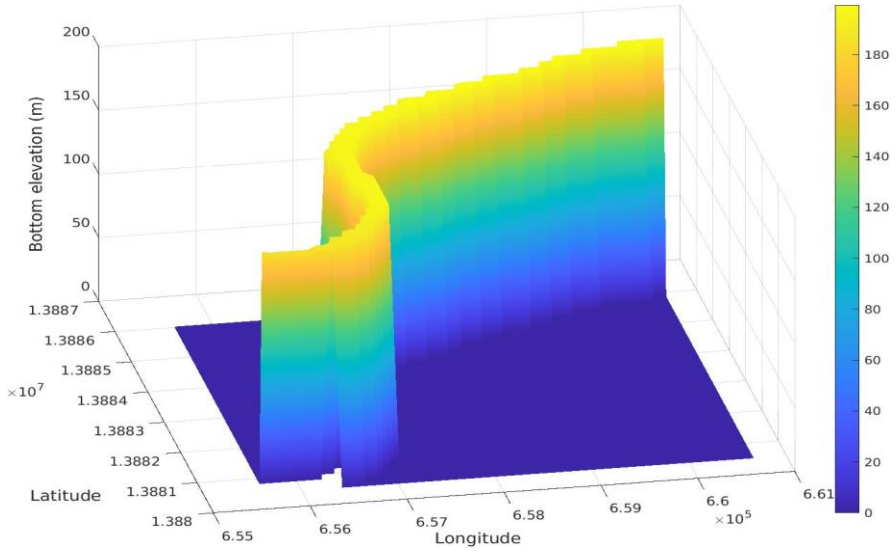

Figure 13 Bathymetry of TN River used in the EFDC model

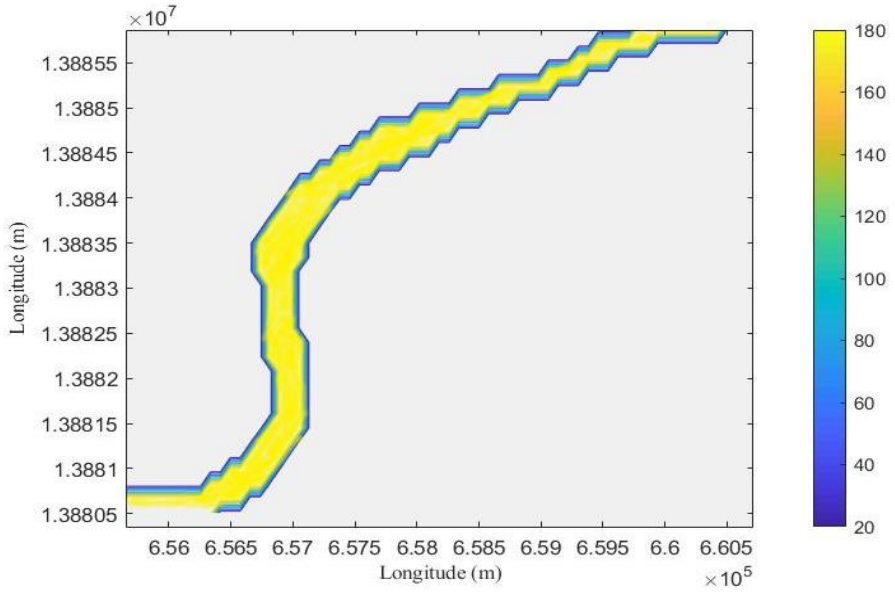

Figure 14 Contour of Bathymetry

Results

The hydrodynamic model was set up for 6 months period starting in January and ending June 2008. To evaluate the results obtained from the EFDC simulation, January 1 to end of June 2008 USGS gage data are compared with model results. This model cell (27,28) is located very closely to the USGS station. To match the simulated flow at cell (27,28) with the observed USGS flow, some adjustments were done in the bathymetry inputs during the calibration. From Figure 15, it is evident that the calibrated model flow results mimic the observed flow values with a high degree of accuracy.

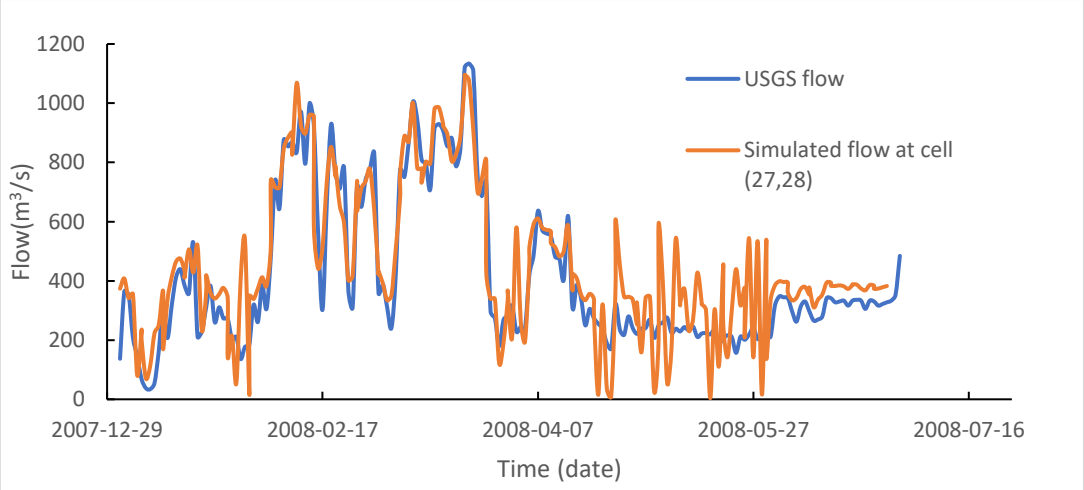

Figure 15 Comparison between USGS and simulated flow at Cell (27,28) from January 2008 to June 2008

Descriptive statistics of observed and simulated flow presented at Table 3 show the similarity of mean, standard deviation, sample variance and other key parameters between observed and simulated flow.

Table 3

Descriptive statistics of Observed and simulated flow

	Observed flow (m^3/s)	Simulated flow (m^3/s)
Mean	415.683	457.411
Standard Error	19.122	18.381
Median	316.400	384.900
Mode	316.400	353.800
Standard Deviation	255.848	245.920
Sample Variance	65458.215	60476.530
Kurtosis	0.021	-0.080
Skewness	1.085	0.611
Range	1099.840	1073.126
Minimum	34.160	3.874
Maximum	1134	1077
Count	179	179

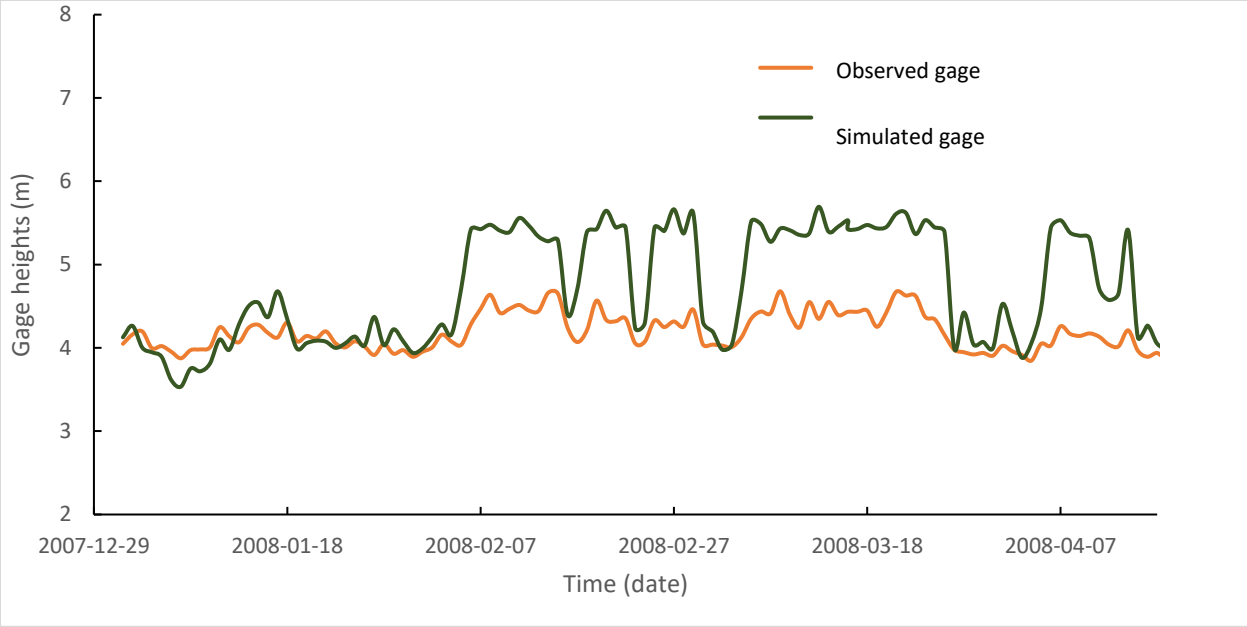

Figure 16 Comparison between USGS and simulated gage heights from January 2008 to June 2008

Open boundary condition used in the downstream of the river was also adjusted simultaneously to match the simulated water level measurements of cell (27,28) with USGS station gage results. From Figure 16, it is evident that the simulated gage heights differ mostly during the February-March period from the observed results. The overestimation of the gage heights during this period may be the result from the inappropriate bathymetry inputs. As a note information about the available bathymetry data is limited. Also, due to the smaller number of grid cells, the shape of the river bathymetry formed a staircase shape.

Table 4 presents the common statistical parameters of observed and simulated gage heights which showed relatively higher differences of standard deviations between observed and simulated gage heights (m).

Table 4

Descriptive statistics of observed and simulated gage heights

	Observed gage (m)	Simulated gage (m)
Mean	4.090	4.491
Standard Error	0.016	0.046
Median	4.015	4.204
Standard Deviation	0.219	0.611
Sample Variance	0.048	0.373
Kurtosis	0.147	-1.001
Skewness	0.994	0.795
Range	0.970	2.159
Minimum	3.710	3.535
Maximum	4.680	5.693
Sum	732.088	803.860
Count	179	179

Figure 17 shows the velocity-time series results which were obtained by dividing the flow values with the cell area. The similarity between the observed and simulated plots thus inferred the good prediction capability for flow velocity generated from the model.

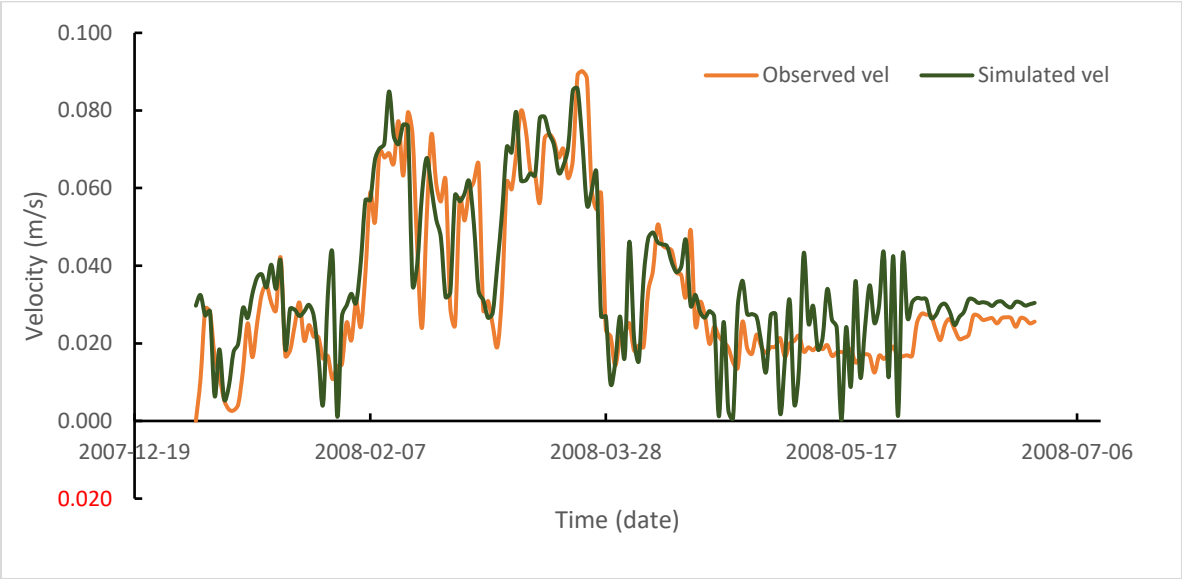

Figure 17 Comparison between USGS and simulated flow velocity at Cell (27,28)

Table 5 presented below shows the statistical parameters of observed and simulated velocity values that indicate the close match among mean, standard deviation, sample variance, and other key statistical parameters. For visualization, vector plot of flow velocity simulated from the test model presented in Figure 18 shows the direction of flow from the upstream to downstream of the river segment.

Table 5

Descriptive statistics of observed and simulated velocity

	Observed Velocity (m/s)		Simulated Velocity (m/s)
Mean	0.033	Mean	0.036
Standard Error	0.002	Standard Error	0.002
Median	0.025	Median	0.031
Standard Deviation	0.020	Standard Deviation	0.020
Sample Variance	0.000	Sample Variance	0.000
Kurtosis	0.021	Kurtosis	-0.080
Skewness	1.085	Skewness	0.611
Range	0.087	Range	0.085
Minimum	0.003	Minimum	0.000
Maximum	0.090	Maximum	0.086
Count	179	Count	179

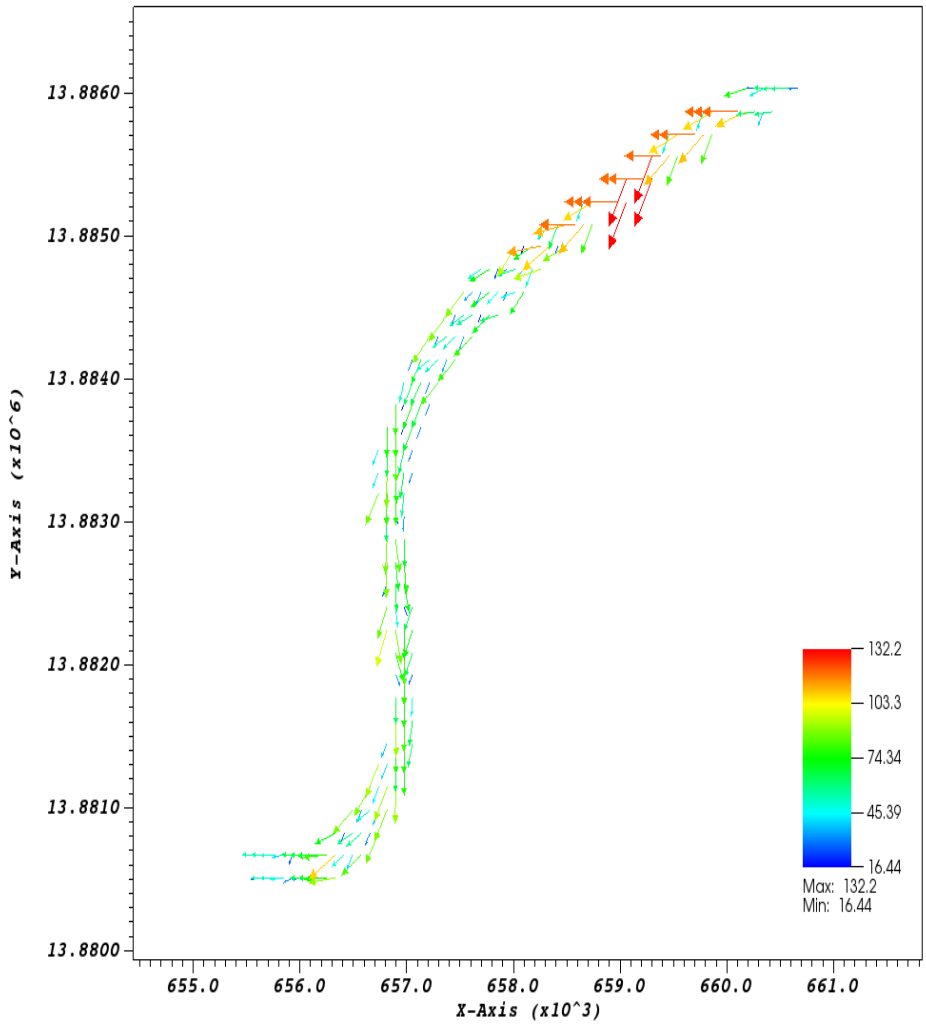

Figure 18 Vector profile of flow in the river section

Calibration

The data required for calibration includes Manning's roughness coefficient for rivers, adjusting the open boundary condition (pressure head or elevation) at the downstream and tweaking the course bathymetry along the cross-section of the river. Calibration is carried out by trial and error method until a reasonable match between observed and modeled water levels,

discharges and velocities is achieved. For this model the Manning's coefficient value was set as 0.025.

For the model validation purpose, key statistical parameters (i.e. Mean Absolute Error, Square Root of the Mean Squared Errors, Normalized Root Mean Square Error, Nash–Sutcliffe Efficiency Coefficient etc.) are calculated and compared between observed and simulated results. The description of the parameters is presented below along with the corresponding results presented in Table 6.

The simplest measure of forecast accuracy is called Mean Absolute Error (MAE) which is the absolute value of the difference between the forecasted value and the actual value. The Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of predictions, without considering their direction.

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|O_n - M_n| \tag{16}$$

Here, O and M are observed and model values whereas, $O_n$and M_n are the nth observed and modeled values respectively.

The RMSE or Square Root of the Mean Squared Error is a good measure of model accuracy that is commonly used to evaluate model performance or differences between observed and predicted values:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(O_n - M_n)^2} \tag{17}$$

The Normalized Root Mean Square Error (NRMSE) relates the RMSE to the observed range of the variable. Thus, the NRMSE can be interpreted as a fraction of the overall range that is typically resolved by the model.

$$NRMSE = \frac{RMSE}{\bar{O}} \tag{18}$$

Where, $\bar{O}$ is the average of observed data.

The Nash–Sutcliffe model efficiency coefficient (NSE) is typically used to assess the predictive power of hydrological models, and is defined as:

$$NSE = 1 - \frac{\sum_{n=1}^{N}(O_n - M_n)^2}{\sum_{n=1}^{N}(O_n - \overline{O}_n)^2} \tag{19}$$

NSE can range from –∞ to 1. An efficiency of 1 (NSE = 1) corresponds to a perfect match between modeled values and observed data; NSE = 0 indicates that the model predictions are as accurate as the mean of the observed data, whereas NSE < 0 occurs when the observed mean is a better predictor than the model. The closer the model efficiency is to 1, the more accurate the model is.

Table 6

Model validation statistics

	Velocity (m/s)	Gage (m)
MAE (m/s)	0.007	0.433
R-Squared	0.819	0.668
RMSE (m/s)	0.009	0.601
NRMSE	0.283	0.147
NSE	0.788	-6.61

From Table 6 it can be inferred that the model is good at capturing the hydrodynamics of the river in terms of flow velocity. The Nash–Sutcliffe model efficiency coefficient (NSE) for velocity is 0.788 which is very close to 1 and indicates the capability of the model to resonate the variability of the flow results during the simulation. On the other hand, the negative NSE value for gage heights suggests the model fails to represent the variability of gage heights and rather provides the simulation results that are often close to the mean values. Besides, the other key statistics of flow velocity also suggest the good prediction quality of the model set up for the Tennessee River. Mean Absolute Error (MAE) for flow velocity is 0.007(m/s) which is low. On the other hand, the MAE for gage height inferring the absolute value of the difference between the simulated value and the actual value is 0.433m which is little bit on the higher side and can be possibly further reduced with fine tuning the bathymetry. Observed and simulated results are used to calculate the statistics (i.e. R-squared error, RMSE etc.) for model fit assessment and are shown in Figure 19.

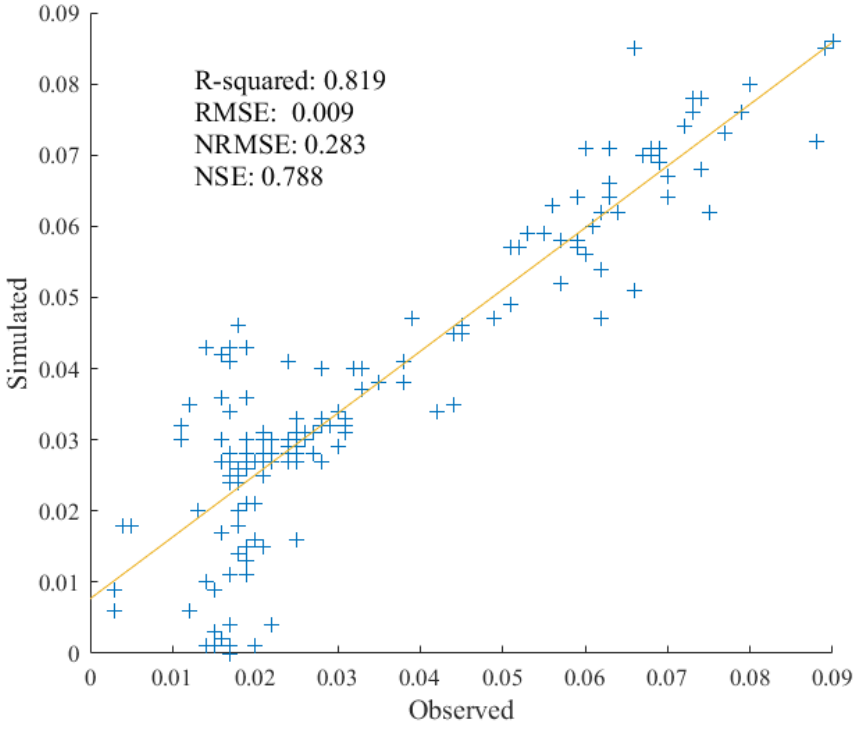

Figure 19 Model fit assessment of simulated and observed flows

CHAPTER V

CONCLUSIONS

A structured quadrilateral grid generator based on triangular elements was developed in MATLAB that successfully generates the cartesian grids and the basic input files required for the hydrodynamic simulation using EFDC. The grid generator was developed considering the complexity of preprocessing steps before setting up the EFDC model in which case it succeeded in making the whole process easier. It will enable a faster EFDC model setup with its user-friendly and reproducible preprocessing utility. The capability of the grid generator is well demonstrated by a real-world example of the Tennessee River hydrodynamic model. The hydrodynamic model set up in this study was run and calibrated from the January to June 2008 period. The comparison of observed and simulated results shows the model's capability to capture the real-world hydrodynamics of the Tennessee River. The calculated gage heights and flows during the simulation period match closely with that of USGS observed results. In this study, the atmospheric time-series data was not considered during the model setup which may not have big impacts on the hydrodynamics for a short segment of the river.

However, further improvement of the Tennessee River model can be achieved by adding the atmospheric time series results along with a detailed bathymetry and a finer cell size that will prepare a perfect hydrodynamics for future water quality analysis. Also, further studies are required to overcome the current limitations of manual adjustments in order to completely

automate grid generator's capability and compatibility for the EFDC model. To improve confidence in the grid generator, as part of future study, it is suggested to compare the EFDC model generated by the newly developed grid generator with that of EFDC model generated by other existing grid generator programs such as EFDC Explorer.

REFERENCE

Ahmad, S., & Simonovic, S. (2000). Comparison of One-Dimensional and Two-Dimensional Hydrodynamic Modeling Approaches for Red River Basin.

Al-Zubaidi, H. A. (2016). 3D Hydrodynamic Model Development and Verification.

Alarcon, V. J., McAnally, W. H., & Pathak, S. (2012). *Comparison of two hydrodynamic models of weeks bay, alabama.* Paper presented at the International Conference on Computational Science and Its Applications.

Atolagbe, B. (2019). *Automatic mesh representation of urban environments.* (M.S. M.S. thesis). University of Tennessee at Chattanooga, Chattanooga (Tenn.). Retrieved from https://scholar.utc.edu/theses/625

Bathi, J., & Roy, S. (2020). Computer Tools for Urban Hydrology and Water Quality Management. In V. G. Veera Gnaneswar Gude, Ramanitharan Kandiah (Ed.), *Sustainable Water: Resources, Management and Challenges* (1st ed.): Nova Science Publishers.

Beletsky, D., & Schwab, D. J. (2001). Modeling circulation and thermal structure in Lake Michigan: Annual cycle and interannual variability. *Journal of Geophysical Research: Oceans, 106*(C9), 19745-19771.

Blumberg, A. F., & Kim, B. N. (2000). Flow balances in St. Andrew Bay revealed through hydrodynamic simulations. *Estuaries, 23*(1), 21-33.

Blumberg, A. F., & Mellor, G. L. (1987). A description of a three-dimensional coastal ocean circulation model. *Three-dimensional coastal ocean models, 4*, 1-16.

Cedillo, P. E. (2015). *Hydrodynamic Modeling of the Green Bay of Lake Michigan Using the Environmental Fluid Dynamics Code.* (Master of Science M.S. thesis). University of Wisconsin-Milwaukee, Milwaukee. Retrieved from https://dc.uwm.edu/etd/1042

Chau, K. W., & Jiang, Y. W. (2001). 3D Numerical Model for Pearl River Estuary. *Journal of Hydraulic Engineering, 127*(1), 72-82. doi:doi:10.1061/(ASCE)0733-9429(2001)127:1(72)

Cunanan, A. M., & Salvacion, J. W. (2016). Hydrodynamic Modeling of Laguna Lake Using Environmental Fluid Dynamics Code. *Int'l Journal of Research in Chemical, Metallurgical and Civil Engg. (IJRCMCE), 3*(1).

Devkota, J., & Fang, X. (2015). Numerical simulation of flow dynamics in a tidal river under various upstream hydrologic conditions. *Hydrological Sciences Journal, 60*(10), 1666-1689.

Hamrick, J. (2002). Theoretical and Computational Aspects of Sediment and Contaminant Transport in the EFDC Model. *Third Draft, Tetra Tech, Inc., Fairfax, VA.*

Hamrick, J. (2007). The environmental fluid dynamics code: User manual US EPA Version 1.01. *Tetra Tech, Inc., Fairfax, VA.*

Hamrick, J. M., & Mills, W. B. (2000). Analysis of water temperatures in Conowingo Pond as influenced by the Peach Bottom atomic power plant thermal discharge. *Environmental Science & Policy, 3*, 197-209.

Ji, Z.-G., Morton, M. R., & Hamrick, J. M. (2000). *Modeling hydrodynamic and sediment processes in Morro Bay.* Paper presented at the Estuarine and coastal modeling.

Jin, K.-R., Ji, Z.-G., & Hamrick, J. H. (2002). Modeling winter circulation in Lake Okeechobee, Florida. *Journal of waterway, port, coastal, and ocean engineering, 128*(3), 114-125.

Liu, L. (2018). Application of a Hydrodynamic and Water Quality Model for Inland Surface Water Systems. In D. Malcangio (Ed.), *Applications in Water Systems Management and Modeling*. London, UK: Intech Open.

Liu, X. (2007). *3D Numerical Modeling of Hydrodynamics and Sediment Transport in Estuaries.* (Ph.D. Ph.D. dissertation). Florida State University, Tallahassee, Florida. Retrieved from http://purl.flvc.org/fsu/fd/FSU_migr_etd-1164

Marvin, J., & Wilson, A. T. (2016). One dimensional, two dimensional and three dimensional hydrodynamic modeling of a Dyked Coastal River in the bay of Fundy. *Journal of Water Management Modeling*.

Panda, R. K., Pramanik, N., & Bala, B. (2010). Simulation of river stage using artificial neural network and MIKE 11 hydrodynamic model. *Computers & Geosciences, 36*(6), 735-745. doi:https://doi.org/10.1016/j.cageo.2009.07.012

Roy, S., Atolagbe, B., Ghasemi, A., & Bathi, J. (2020). *A MATLAB based Grid Generation Tool for Hydrodynamic Model Set-up*. Paper presented at the EWRI Watershed Management Conference, Henderson, Nevada.

Schubert, J. E., Sanders, B. F., Smith, M. J., & Wright, N. G. (2008). Unstructured mesh generation and landcover-based resistance for hydrodynamic modeling of urban flooding. *Advances in Water Resources, 31*(12), 1603-1621. doi:https://doi.org/10.1016/j.advwatres.2008.07.012

Shewchuk, J. R. (1996). Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied computational geometry towards geometric engineering* (pp. 203-222): Springer.

Sucsy, P., Carter, E., Christian, D., Cullum, M., Park, K., Stewart, J., & Zhang, Y. (2002). RIVER HYDRODYNAMICS CALIBRATION. In *St. Johns River Water Supply Impact Study*. St. Johns River Water Management District, Palatka, FL. 343

Tetra Tech Inc. (2002). *VOGG: A Visual Orthogonal Grid Generation Tool for Hydrodynamic and Water Quality Modeling*. Retrieved from

Wang, Y., Jiang, Y., Liao, W., Gao, P., Huang, X., Wang, H., . . . Lei, X. (2014). 3-D hydro-environmental simulation of Miyun reservoir, Beijin. *Journal of Hydro-environment Research, 8*(4), 383-395. doi:https://doi.org/10.1016/j.jher.2013.09.002

Xiong, Y. (2010). *Coupling sediment transport and water quality models.* (Ph.D. Ph.D. dissertation). Mississippi State University, Mississippi.

APPENDIX A

CARD IMAGES AND INPUT FILES FOR EFDC

```
C7 TIME-RELATED INTEGER PARAMETERS
*
* NTC:    NUMBER OF REFERENCE TIME PERIODS IN RUN
* NTSPTC:  NUMBER OF TIME STEPS PER REFERENCE TIME PERIOD
* NLTC:    NUMBER OF LINEARIZED REFERENCE TIME PERIODS
* NLTC:    NUMBER OF TRANSITION REF TIME PERIODS TO FULLY NONLINEAR
* NTCPP:   NUMBER OF REFERENCE TIME PERIODS BETWEEN FULL PRINTED OUTPUT
*         TO FILE EFDC.OUT
* NTSTBC:  NUMBER OF REFERENCE TIME PERIODS BETWEEN TWO TIME LEVEL
*         TRAPEZOIDAL CORRECTION TIME STEP, ** MASS BALANCE PRINT INTERVAL **
* NTCNB:   NUMBER OF REFERENCE TIME PERIODS WITH NO BUOYANCY FORCING
* NTCVB:   NUMBER OF REFERENCE TIME PERIODS WITH VARIABLE BUOYANCY FORCING
* NTSMMT:  NUMBER OF REFERENCE TIME TO AVERAGE OVER TO OBTAIN
*         RESIDUAL OR MEAN MASS TRANSPORT VARIABLES
* NFLTMT:  USE 1 (FOR RESEARCH PURPOSES)
* NDRYSTP: MIN NO. OF TIME STEPS A CELL REMAINS DRY AFTER INITIAL DRYING
*         -NDRYSTP FOR ISDRY=-99 TO ACTIVATE WASTING WATER IN DRY CELLS
C7 NTC NTSPTC  NLTC NTTC NTCPP NTSTBC NTCNB NTCVB NTSMMT NFLTMT NDRYSTP
  180   14400   0    0    800   4      0     2     720    1      16
--------------------------------------------------------------------------------



C9  SPACE-RELATED AND SMOOTHING PARAMETERS
*
*     IC:      NUMBER OF CELLS IN I DIRECTION
*     JC:      NUMBER OF CELLS IN J DIRECTION
*     LC:      NUMBER OF ACTIVE CELLS IN HORIZONTAL + 2
*     LVC:     NUMBER OF VARIABLE SIZE HORIZONTAL CELLS
*     ISCO:    1 FOR CURVILINEAR-ORTHOGONAL GRID (LVC=LC-2)
*     NDM:     NUMBER OF DOMAINS FOR HORIZONTAL DOMAIN DECOMPOSITION
*             ( NDM=1, FOR MODEL EXECUTION ON A SINGLE PROCESSOR SYSTEM OR
*               NDM=MM*NCPUS, WHERE MM IS AN INTEGER AND NCPUS IS THE NUMBER
*               OF AVAILABLE CPU's FOR MODEL EXECUTION ON A PARALLEL
*               MULTIPLE PROCESSOR SYSTEM )
*     LDW:     NUMBER OF WATER CELLS PER DOMAIN
*             ( LDW=(LC-2)/NDM, FOR MULTIPE VECTOR PROCESSORS, LWD MUST BE
*               AN INTEGER MULTIPLE OF THE VECTOR LENGTH OR STRIDE NVEC
*               THUS CONSTRAINING LC-2 TO BE AN INTEGER MULTIPLE OF NVEC )
*     ISMASK:  1 FOR MASKING WATER CELL TO LAND OR ADDING THIN BARRIERS
*             USING INFORMATION IN FILE mask.inp
*    ISPGNS:  1 FOR IMPLEMENTING A PERIODIC GRID IN COMP N-S DIRECTION OR
*             CONNECTING ARBITRARY CELLS USING INFORMATION IN FILE mappgns.inp
*     NSHMAX:  NUMBER OF DEPTH SMOOTHING PASSES
*     NSBMAX:  NUMBER OF INITIAL SALINITY FIELD SMOOTHING PASSES
*     WSMH:    DEPTH SMOOTHING WEIGHT
*     WSMB:    SALINITY SMOOTHING WEIGHT
*
C9  IC  JC  LC   LVC  ISCO NDM LDW  ISMASK ISPGNS NSHMX NSBMX   WSMH   WSMB
   70  42  193  191  0    1   191  0      0      0      0      0      0
--------------------------------------------------------------------------------
```

```
C9A  VERTICAL SPACE-RELATED PARAMETERS
*
*     KC:      NUMBER OF VERTICAL LAYERS
*    KSIG:     NUMBER OF VERTICAL LAYERS IN SIGMA REGION FOR IGRIDV = 1
* ISETGVC:    0 READ BOTTOM LAYER ID FROM GVCLAYER.INP
*             1 AUTOMATICALLY SET BOTTOM LAYER ID USING SELVREF, SELVREF
*                                 AND BELV (IN DXDY.INP) AND WRITE RESULTS TO GVCLAYER.OUT
* SELVREF:     REFERENCE SURFACE ELEVATION IN RESCALED HEIGHT REGION (METERS)
* BELVREF:     REFERENCE (MINIMUM) BOTTOM ELEVATION IN RESCALED HEIGHT REGION
* ISGVCCK:    0 NORMAL SETTING (OPTION 1 USED FOR DEBUGGING SIGMA/GVC COMPARE)
*             1 USE MULTI-LAYER BOTTOM FRICTION FOR SINGLE LAYER SIGMA
*
C9A  KC  KSIG  ISETGVC  SELVREF  BELVREF  ISGVCCK
   2  1   1       -69.5     -84.75     0
-------------------------------------------------------------------------------
C10  LAYER THICKNESS IN VERTICAL
*
*       K:  LAYER NUMBER, K=1,KC
*      DZC:  DIMENSIONLESS LAYER THICKNESS (THICKNESSES MUST SUM TO 1.0)
*           FOR IGRIDV=1, THE TOP KSIG LAYERS ARE PRESENT IN BOTH THE
*           SIGMA AND RESCALED HEIGHT REGIONS
*
C10    K   DZC
      1        0.5
      2        0.5
-------------------------------------------------------------------------------


C16 SURFACE ELEVATION OR PRESSURE BOUNDARY CONDITION PARAMETERS
*
*     NPBS:    NUMBER OF SURFACE ELEVATION OR PRESSURE BOUNDARY CONDITIONS
*              CELLS ON SOUTH OPEN BOUNDARIES
*     NPBW:    NUMBER OF SURFACE ELEVATION OR PRESSURE BOUNDARY CONDITIONS
*              CELLS ON WEST OPEN BOUNDARIES
*     NPBE:    NUMBER OF SURFACE ELEVATION OR PRESSURE BOUNDARY CONDITIONS
*              CELLS ON EAST OPEN BOUNDARIES
*     NPBN:    NUMBER OF SURFACE ELEVATION OR PRESSURE BOUNDARY CONDITIONS
*              CELLS ON NORTH OPEN BOUNDARIES
*     NPFOR:   NUMBER OF HARMONIC FORCINGS
*     NPFORT:  FORCING TYPE, 0 = CONSTANT, 1 = LINEAR, 2 = QUADRATIC VARIATION
*     NPSER:   NUMBER OF TIME SERIES FORCINGS
*     PDGINIT: ADD THIS CONSTANT ADJUSTMENT GLOBALLY TO THE SURFACE ELEVATION
*
C16  NPBS NPBW NPBE NPBN NPFOR NPFORT NPSER PDGINIT
    0   2    0    0    0     0    1     0.0     0
-------------------------------------------------------------------------------
```

```
C19 PERIODIC FORCING (TIDAL) SURF ELEV OR PRESSURE ON WEST OPEN BOUNDARIES
*
*     IPBW:      SEE CARD 19
*     JPBW:
*     ISPBW:
*     NPFORW:
*     NPSERW:
*     TPCOORDW:
*
C19 IPBW JPBW   ISPBW NPFORW  NPSERW  TPCOORDW
    5    5    0    0       1      0
    5    6    0    0       1      0
-------------------------------------------------------------------------------


C23 VELOCITY, VOLUME SOURCE/SINK, FLOW CONTROL, AND WITHDRAWAL/RETURN DATA
*
* NQSIJ:  NUMBER OF CONSTANT AND/OR TIME SERIES SPECIFIED SOURCE/SINK
*         LOCATIONS (RIVER INFLOWS, ETC)
* NQJPIJ: NUMBER OF CONSTANT AND/OR TIME SERIES SPECIFIED SOURCE
*         LOCATIONS TREATED AS JETS/PLUMES
* NQSER:  NUMBER OF VOLUME SOURCE/SINK TIME SERIES
* NQCTL:  NUMBER OF PRESSURE CONTROLLED WITHDRAWAL/RETURN PAIRS
* NQCTLT: NUMBER OF PRESSURE CONTROLLED WITHDRAWAL/RETURN TABLES
* NQWR:   NUMBER OF CONSTANT OR TIME SERIES SPECIFIED WITHDRAWAL/RETURN
*         PAIRS
* NQWRSR: NUMBER OF TIME SERIES SPECIFYING WITHDRAWAL, RETURN AND
*         CONCENTRATION RISE SERIES
* ISDIQ:  SET TO 1 TO WRITE DIAGNOSTIC FILE, DIAQ.OUT
*
C23  NQSIJ  NQJPIJ  NQSER  NQCTL  NQCTLT  NQWR  NQWRSR  ISDIQ
     4    0      2     0     0     0    0      0
-------------------------------------------------------------------------------
```

```
C24 VOLUMETRIC SOURCE/SINK LOCATIONS, MAGNITUDES, AND CONCENTRATION SERIES
*
*  IQS:     I CELL INDEX OF VOLUME SOURCE/SINK
*  JQS:     J CELL INDEX OF VOLUME SOURCE/SINK
*  QSSE:    CONSTANT INFLOW/OUTFLOW RATE IN M*M*M/S
*  NQSMUL:  MULTIPLIER SWITCH FOR CONSTANT AND TIME SERIES VOL S/S
*         = 0  MULT BY 1. FOR NORMAL IN/OUTFLOW (L*L*L/T)
*         = 1  MULT BY DY FOR LATERAL IN/OUTFLOW (L*L/T) ON U FACE
*         = 2  MULT BY DX FOR LATERAL IN/OUTFLOW (L*L/T) ON V FACE
*         = 3  MULT BY DX+DY FOR LATERAL IN/OUTFLOW (L*L/T) ON U&V FACES
*  NQSMFF:  IF NON ZERO ACCOUNT FOR VOL S/S MOMENTUM FLUX
*         = 1  MOMENTUM FLUX ON NEG U FACE
*         = 2  MOMENTUM FLUX ON NEG V FACE
*         = 3  MOMENTUM FLUX ON POS U FACE
*         = 4  MOMENTUM FLUX ON POS V FACE
*  NQSERQ:  ID NUMBER OF ASSOCIATED VOLUME FLOW TIME SERIES
*  NSSERQ:  ID NUMBER OF ASSOCIATED SALINITY TIME SERIES
*  NTSERQ:  ID NUMBER OF ASSOCIATED TEMPERATURE TIME SERIES
*  NDSERQ:  ID NUMBER OF ASSOCIATED DYE CONC TIME SERIES
*  NSFSERQ: ID NUMBER OF ASSOCIATED SHELL FISH LARVAE RELEASE TIME SERIES
*  NTXSERQ: ID NUMBER OF ASSOCIATED TOXIC CONTAMINANT CONC TIME SERIES
*  NSDSERQ: ID NUMBER OF ASSOCIATED COHESIVE SEDIMENT CONC TIME SERIES
*  NSNSERQ: ID NUMBER OF ASSOCIATED NON-COHESIVE SED CONC TIME SERIES
*  QSFACTOR: FACTOR FOR FLOW SERIES
*
C24 IQS   JQS    QSSE    NQSMUL NQSMFF NQSERQ NS- NT- ND- NSF- NTX- NSD- NSN- QSFACTOR
   68     40    1.0       0     0      1      0   0   0   0    0    0    0    1.0   inflow1
   65     39    1.0       0     0      2      0   0   0   0    0    0    0    1.0   inflow2
   68     40   -1.0       0     0      0      0   0   0   0    0    0    0    1.0  outflow1
   65     39   -1.0       0     0      0      0   0   0   0    0    0    0    1.0   outflow2
```

```
C87 CONTROLS FOR WRITING TO TIME SERIES FILES
*
*  ILTS:    I CELL INDEX
*  JLTS:    J CELL INDEX
*  NTSSSS:  WRITE SCENARIO FOR THIS LOCATION
*  MTSP:  1 FOR TIME SERIES OF SURFACE ELEVATION
*  MTSC:  1 FOR TIME SERIES OF TRANSPORTED CONCENTRATION VARIABLES
*  MTSA:  1 FOR TIME SERIES OF EDDY VISCOSITY AND DIFFUSIVITY
*  MTSUE: 1 FOR TIME SERIES OF EXTERNAL MODE HORIZONTAL VELOCITY
*  MTSUT: 1 FOR TIME SERIES OF EXTERNAL MODE HORIZONTAL TRANSPORT
*  MTSU:  1 FOR TIME SERIES OF HORIZONTAL VELOCITY IN EVERY LAYER
*  MTSQE: 1 FOR TIME SERIES OF NET EXTERNAL MODE VOLUME SOURCE/SINK
*  MTSQ:  1 FOR TIME SERIES OF NET EXTERNAL MODE VOLUME SOURCE/SINK
*  CLTS:    LOCATION AS A CHARACTER VARIABLE
*
C87 ILTS JLTS NTSSSS MTSP MTSC MTSA MTSUE MTSUT MTSU MTSQE MTSQ CLTS
   20   18    1      1    1    0    0     1    0    0    0   'xsec1'
   21   18    1      1    1    0    0     1    0    0    0   'xsec1'
   22   18    1      1    1    0    0     1    0    0    0   'xsec1'
   24   28    1      1    1    0    0     1    0    0    0   'USGS Sec'
   25   28    1      1    1    0    0     1    0    0    0   'USGS Sec'
   26   28    1      1    1    0    0     1    0    0    0   'USGS Sec'
   27   28    1      1    1    0    0     1    0    0    0   'USGS Sec'
   29   31    1      1    1    0    0     1    0    0    0   'xsec1'
   30   31    1      1    1    0    0     1    0    0    0   'xsec2'
   31   31    1      1    1    0    0     1    0    0    0   'xsec2'
   32   31    1      1    1    0    0     1    0    0    0   'xsec2'
   33   31    1      1    1    0    0     1    0    0    0   'xsec2'
   34   31    1      1    1    0    0     1    0    0    0   'xsec2'
   35   31    1      1    1    0    0     1    0    0    0   'xsec2'
   36   31    1      1    1    0    0     1    0    0    0   'xsec2'
   31   31    1      1    1    0    0     1    0    0    0   'xsec2'
    5    5    1      1    1    0    0     1    0    0    0   'open bndry'
    5    6    1      1    1    0    0     1    0    0    0   'open bndry'
```

```
C   qser.inp file, in free format across line, repeats nqser times
C
C   ISTYP   MQSER(NS)   TCQSER(NS)  TAQSER(NS)  RMULADJ(NS)  ADDADJ(NS)
C
C   if istyp.eq.1 then read depth weights and single value of QSER
c
C   (WKQ(K),K=1,KC)
c
C   TQSER(M,NS)  QSER(M,1,NS)   !(mqser(ns) pairs for ns=1,nqer series)
c
C   else read a value of qser for each layer
c
C   TQSER(M,NS)   (QSER(M,K,NS),K=1,KC)  !(mqser(ns) pairs)
C
        1   182   86400.   0.    1.0   0.    0   ! nqser =    1   inflow 1
     0.500 0.500
     1       136.92
     2       364
     3       352.8
     4       208.88
     5       145.32
     6       68.6
     7       37.8
     8       34.16
     9       54.32
     10      164.08
     11      316.4
     12      206.64
     13      316.4
     14      411.6
     15      439.6
     16      383.6
     17      358.4
     18      529.2
     19      211.12
     20      228.48
     21      310.8
     22      383.6
     23      260.4
     24      310.8
     25      273.28
     26      271.6
```

```
C ** , pser.inp Time Series FILE
C **   REPEATS NPSER TIMES
C **   ITYP(NS)  MPSER(NS)   TCPSER(NS)  TAPSER(NS)  RMULADJ(NS)  ADDADJ(NS) PSERZDF(NS)  INTPSER(NS)
C **
C **    ITYP = 0-SINGLE VALUE WITH 1 HEADER LINE,  1-HIGH LOW SEPARATED WITH 2 HEADER LINES
C ** PSERZDF = OFFSET (M)
C **
C **
C **   THE FOLLOWING REPEATED  MPSER(NS) TIMES
C **   TPSER(M,NS)  PSER(M,1,NS) [ PSERS(M,1,NS) ]
       0      6   86400      0      1      0      0 !  constant downstream head
     0.00  179.7
     30    179
     60    179.8
     90    179.08
     120   179.08
     180   179.28


       0      6   86400      0      1      0      0 !  constant downstream head
     0.00    179.7
     30    179
     60    179.8
     90    179.08
     120   179.08
     180   179.28
```

APPENDIX B

GRID GENERATOR ALGORITHM

Define (nx, ny) as number of quad cells in both directions.
Set file name for input file for raw city data i.e. fname1
Set file name for output file of grids generated i.e. fname2
[grd] ← algorithm 15
Store grid data in $fname2$
Output: $fname2$
End

This function generates the grid data for the Waterscape mesh
Input: nx, ny,N_{seg},$XYZRaw$, $EdgesRaw$
Output: $cells$

Step 1: Get triangular grids for Waterscape
Extract coordinates and connectivity into arrays XYZ, $Edges$ respectively.
for i in N_{seg} **do**
    Store $XYZ_{seg}, Edges_{seg}$ in $XYZ, Edges$ ▷ Input to algorithm 19
end for
if $N_{seg} = 1$ **then**
    Island doesn't exists
    $Holes \leftarrow null$
else
    Let $i_{seg} \leftarrow 1$
    for i_{seg} in $(N_{seg} - 1)$ **do**
        Extract $XYZ_{seg}, Edges_{seg}$ ▷ Input to algorithm18
        $XYZ_{island}, ICON_{island} \leftarrow$ algorithm 18
        Pick any triangle
        $r \leftarrow$ random cell (or row) number in $ICON_{island}$
        $RndmCell \leftarrow ICON_{island}(r)$ ▷ RndmCell is a 1x3 vector
        $pt_i = RndmCell(i)$
        $V_j \leftarrow XYZ_{island}(pt_j, 1 : end); j = 1 : 3$ ▷ V is 1x3 vector of coordinates of the vertices pt_i.
        $x_k = V_j(k, 1),\ k = 1 : 3$
        $y_k = V_j(k, 2),\ k = 1 : 3$
        $Holes(i) \leftarrow \frac{\sum_{k=1}^{3} x_k, y_k}{3}$ ▷ Input to algorithm 19
    end for
end if
XYZ_{tri}, $ICON_{tri} \leftarrow$ algorithm 19

Step2: Define quad cells beyond the Waterscape domain

Extend two distant corner points defined by XYZ_{tri} to $P1, P2$ by a desired length

Let $x1 = P1_x$, $y1 = P1_y$,$x2 = P2_x$, $y2 = P2_y$,

Calculate cell size: $dx, dy \leftarrow \dfrac{x2, y2 - x1, y1}{nx, ny}$

Hold $x1_{init} \leftarrow x1$

Hold $y1_{init} \leftarrow y1$

Get coordinates of grid cells

initialize $inode, jnode \leftarrow 1$

initialize $jj \leftarrow 1$

for $jnode$ in $(ny + 1)$ **do**

    for $inode$ in $(nx + 1)$ **do**

        $x(inode, jnode) = x1 + dx$;

        $y(inode, jnode) = y1$;

        Check point x, y in $ICON_{tri}$; Add tags $T = 1$ or 0

        $T(inode, jnode) \leftarrow$ algorithm 17

        $index(inode, jnode) = jj$

        $jj = jj + 1$

        $x1 = x1 + dx$

    end for

    $x1 = x1_{init}$

    $y1 = y1 + dy$

end for

Calculate cell values

Initialize $icell, jcell \leftarrow 1$

for $jcell$ in ny **do**

    for $icell$ in nx **do**

        $cells\{icell, jcell\}.corners.x = x(icell, jcell, x(icell + 1, jcell), x(icell + 1, jcell + 1), x(icell, jcell + 1)$

        $cells\{icell, jcell\}.corners.y = y(icell, jcell, y(icell + 1, jcell), y(icell + 1, jcell + 1), y(icell, jcell + 1)$

        $cells\{icell, jcell\}.centroid = \bar{x}, \bar{y}$

        $cells\{icell, jcell\}.dxdy = dx, dy$

        $cells\{icell, jcell\}.tags = T(icell, jcell), T(icell + 1, jcell, T(icell + 1, jcell + 1), T(icell, jcell + 1)$

    end for

end for

Step3: Write EFDC File

$cells \leftarrow$ algorithm 20

Output grd

End

1: This function reads the input file containing raw data of the segments
2: Input: $filename.dat$
3: Output: XYZ, $Edges$, N_{seg}

4: Open $filename$
5: Read Number of segments N_{seg}
6: Initialize $jj \leftarrow 1$, $jj_{init} \leftarrow 1$
7: $i_{seg} \leftarrow 1$; $inode \leftarrow 1$
8: **for** i_{seg} in N_{seg} **do**
9: Read number of points in this segment: $Nodes$
10: **for** $inode$ in $Nodes$ **do**
11: Read coordinates of $inode$: x_i, y_i, z_i
12: $XYZTemp(inode) \leftarrow x_{inode}, y_{inode}, z_{inode}$
13: $Edge1 \leftarrow jj$
14: **if** $inode = Nodes$ **then**
15: $Edge2 \leftarrow jj_{init}$
16: **else**
17: $Edge2 \leftarrow jj + 1$
18: **end if**
19: $EdgeTemp(inode) \leftarrow Edge1 : Edge2$
20: $jj = jj + 1$
21: **end for**
22: $jj_{init} \leftarrow jj_{init} + Nodes$
23: $Edges\{i_{seg}\} \leftarrow EdgeTemp$
24: $XYZ\{i_{seg}\} \leftarrow XYZTemp$
25: **end for**
26: Close $filename$
27: End

1: This algorithm checks point $XYpt_{x,y}$ if it falls within any, or none, of the triangular cells of the river mesh and add tags of 1 or 0 respectively..
2: Input: $XYZtri, ICONtri, XYpt$
3: Output: tag

4: Initialize $tag \leftarrow 0$, $itri \leftarrow 1$
5: $N_{elem} \leftarrow$ number of elements (or rows) in $ICONtri$
6: **for** $itri$ in N_{elem} **do**
7: **if** $tag = 1$ **then**
8: Skip
9: **end if**
10: $Row = ICONtri(itri, :)$
11: $pt_i = RndmCell(i)$
12: $V_j \leftarrow XYZtri(pt_j, 1 : end)$ ▷ V is $1x3$ vector of coordinates of the three vertices in the random cell.
13: $x_k = V_j(k, 1) : k = 1 : 3$
14: $y_k = V_j(k, 2) : k = 1 : 3$

$$A = \begin{bmatrix} (x1 - x3) & (x2 - x3) \\ (y1 - y3 & (y2 - y3) \end{bmatrix}$$

$$b = \begin{bmatrix} (x - x3) \\ (y - y3)) \end{bmatrix}$$

15: $Out = A \backslash b$
16: $Out = A\ b$
17: $r = Out(1);\ s = Out(2)$
18: **if** $r \geq 0$ & $s \geq 0$ & $(r + s) \leq 1$ **then**
19: $tag \leftarrow 1$ ▷ Point falls in one of the triangles
20: **else**
21: $tag \leftarrow 1$ ▷ Point doesn't fall in any of the triangles.
22: **end if**
23: **end for**
24: End

1: This function generates temporary triangular meshes for the Waterscape
2: Input: XYZ, $Edges$
3: Output: XY, $ICON$

4: $filename \leftarrow$ algorithm 13 ▷ Rewrite edge segment into Triangle input file
5: Set $Holes \leftarrow null$
6: $XY, ICON \leftarrow$ algorithm12 ▷ Includes path to Triangle.
7: Set $H \leftarrow zero$
8: **for** All XY **do**
9: $XYZ \leftarrow$ Add H to $(XY)_i$
10: **end for**
11: End

1: This function generates temporary grids for dry segments (or islands) such that can be used for establishing holes for the island.
2: Input: XYZ, $Edges$
3: Output: XY, $ICON$

4: $filename \leftarrow$ algorithm 13 ▷ Rewrite edge segment into Triangle input file
5: Set $Holes \leftarrow null$
6: $XY, ICON \leftarrow$ algorithm 12 ▷ Includes path to Triangle.
7: End

```
This function assigns tags to all cells tags according to the EFDC requirements
Input: cells, nx, ny
Output: cells

Initialize i, j ← 1
for j in ny do
    for i in nx do
        tags ← cells{i, j}.tags
        if tags = 1111 then
            cells{i, j}.efdctags = 5
        else if tags = 1101 then
            cells{i, j}.efdctags = 1
        else if tags = 1011 then
            cells{i, j}.efdctags = 2
        else if tags = 1011 then
            cells{i, j}.efdctags = 3
        else if tags = 1110 then
            cells{i, j}.efdctags = 4
        else if tags = 0000 then
            cells{i, j}.efdctags = 0
        else
            cells{i, j}.efdctags = 9
        end if
    end for
end for
End
```

VITA

Shuvashish Roy was born in Panchagarh, Bangladesh, to Amrita Kumar Roy and Mukta Rani Roy on June 22, 1992. He is the oldest of two siblings. He obtained his Bachelor of Science in Civil Engineering from Bangladesh University of Engineering and Technology in September 2015. Shuvashish received a full funded Research Assistantship during his Master of Science program at the University of Tennessee at Chattanooga (UTC) in Civil Engineering. He graduated from UTC in May 2020.